Causality in Pure Quantum Computation with Quantum Control

Kengo Hirata
University of Edinburgh, Edinburgh, United Kingdom
Kyoto University, Kyoto, Japan

Takeshi Tsukada
Chiba University, Chiba, Japan

Abstract

Indefinite causal order is a characteristic phenomenon in quantum computation, with examples including the quantum SWITCH and the OCB process. Not all such processes are believed to be physically realizable: while some implementations of the quantum SWITCH have been proposed, the OCB process is suspected to be unrealizable. This difference in realizability is commonly attributed to constraints imposed by physical causality.

This paper studies such a causality issue in a higher-order setting, proposing a typed lambda calculus with quantum control and its categorical semantics. Our calculus extends pure quantum computation with higher-order functions and quantum conditional branching, and it is equipped with a type system based on intuitionistic BV logic to enforce causality. We also present a novel model that is closely related to the Caus construction, by which we prove that some physically-unrealizable processes are not definable in our language.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Type theory; Theory of computation → Linear logic; Theory of computation → Categorical semantics

Keywords and phrases Quantum computing, supermap, categorical model, linear logic, BV logic, programming language, type system

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.57

Acknowledgements This work was supported by JST CREST, Japan, Grant Number JPMJCR25I5.

1 Introduction

Quantum computation is a computational paradigm that exploits the principles of quantum mechanics, and it can efficiently solve certain problems that are not efficiently solvable by any known classical algorithm. Naturally, the scope of quantum computation is constrained by the operations that quantum mechanics allows. In the standard formalism, a physical system is represented by a Hilbert space, and the physically admissible transformations between Hilbert spaces are well understood. Concretely, depending on the class of operations that one allows, physical processes are modeled by unitary transformations, isometries, or quantum channels. Furthermore, every physically realizable operation admits a description in terms of quantum circuits.

However, the situation changes dramatically once we move to second-order operators, *i.e.*, transformations that take quantum channels as inputs and return a new quantum channel. Such transformations are called *supermaps*. As we shall see below, this setting exhibits novel and intriguing phenomena that do not arise at first order, but many fundamental questions remain open, including which transformations are physically realizable and how such operations can be described in an appropriate language.

One such phenomenon is *indefinite causal order*, of which a prominent example is the *quantum SWITCH* [17]. The quantum SWITCH takes two quantum channels A and B as

inputs and behaves as $A \circ B$ or $B \circ A$ depending on a control qubit. In particular, when the control qubit is in a superposition of $|0\rangle$ and $|1\rangle$, the quantum switch performs a "mixture" or "superposition" of $A \circ B$ and $B \circ A$, hence "which of A and B was executed first" is not well-defined. There is also research that investigates computational advantages that exploit this property [5, 16, 20, 21, 36, 37, 53]. Another instance of this phenomenon is the supermap introduced by Oreshkov, Costa and Brukner [41], which we call the *OCB process*. Its remarkable feature is the violation of the *causal inequality* [13, 41, 46, 55], which is unviolatable under classical causality.

Although the quantum SWITCH [17] and the OCB process [41] are both interesting supermaps exhibiting indefinite causal order, their physical realizability is in different situations. For the quantum SWITCH, several implementation proposals and experimental results have been reported [23, 39, 45, 57]. In contrast, to the best of our knowledge, no implementation or experimental realization of the OCB process has been given, and it is rather suspected to be unrealizable [6, 46].

The precise boundary between realizable and unrealizable supermaps is of significant interest, yet remains unclear. To explore the boundary, or as candidates of the boundary, various classes of supermaps have been proposed.

One of the most fundamental is the class of *pure supermaps* [7, 6]. This is the class of supermaps that, when the inputs are "pure" quantum operations, *i.e.*, those involving neither measurement nor discarding, similarly return a pure operation. The purity distinguishes between the quantum SWITCH and the OCB process: the quantum SWITCH is pure, but the OCB process is not. Several other classes have been proposed [22, 42, 55] as subclasses of *purifiable supermaps*, *i.e.*, pure supermaps followed by measurements, making purity a useful baseline notion.

Kissinger and Uijlen [35] and Simmons and Kissinger [51, 52] studied supermaps from categorical and logical viewpoints. Kissinger and Uijlen [35] have introduced the *Caus construction*, which yields a category that can model higher-order causal structure in probabilistic and quantum theories. Simmons and Kissinger [51, 52] showed that the resulting category provides a model for causality-aware extensions of linear logic, namely BV-logic [25] and pomset-logic [47]. This provides a striking link between notions of causal structure in physics and in logic. Hefford and Wilson [28] have extended their work and created a model which characterizes pure-reversible quantum computation[1] and proved that their model can also interpret BV-logic, strengthening the relation of causal structure in the two field.

The above research can be regarded as giving a declarative description of realizable supermaps. The complementary direction is to ask what kinds of constructions are allowed to build realizable supermaps. In terms of programming language, this amounts to asking what kinds of language constructs can be introduced to a programming language for supermaps.

Quantum control is a construction that has recently attracted attention. It is a programming principle for constructing pure higher-order processes [55], allowing a program to form coherent superpositions of different executions. The quantum SWITCH is the canonical example of this idea, creating a superposition of different applications of input operations. Importantly, this mechanism remains causally constrained: it can generate indefinite causal order, but by itself cannot violate causal inequalities [46, 55], unlike processes such as the OCB process. Thus quantum control occupies an intermediate regime: it goes beyond fixed-order higher-order programs while still falling short of processes that violates causal

[1] Here, by pure-reversible, we mean the subclass of quantum computation where all first order functions are unitaries and higher order functions preserves them.

Figure 1 Causality violation in naive higher-order quantum control in (∗∗). (Left) The function f acts as a NOT gate controlled by the qubit x. Each f_{in} and f_{out} represents the input and output of f. (Right) Applying f to the control qubit itself creates a closed loop.

inequalities. This makes quantum control a natural test case for understanding causality in pure higher-order quantum computation.

The above-mentioned causal models capture causal constraints on higher-order processes, but they cannot interpret quantum control at least in an evident way. This paper studies the causality in the coexistence of quantum control and higher-order functions.

New Causality Issue in Higher-order Programs. The first contribution of this paper is to identify a new causality issue arising from the combination of quantum control and higher-order functions. Quantum control, exemplified by the controlled-U gate (that applies U when the control qubit is $|1\rangle$ and acts as the identity when the control qubit is $|0\rangle$), is a fundamental ingredient of quantum computation and also underlies the quantum SWITCH. However, a quantum-controlled higher-order function gives rise to nontrivial causal problems.

We use the notation $\mathsf{qif}\, M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2$ for the syntax of quantum control in our language to represent positively controlled N_1 and negatively controlled N_2 by a qubit M in analogy with classical conditional $\mathsf{if}\, M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2$. For example, the term $\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\, y\ \mathsf{else}\ y$ represents controlled-NOT gate applied to qubits x and y, where X is the NOT gate that flips $|0\rangle$ to $|1\rangle$ and vice versa. In contrast to the classical case, $\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\, y\ \mathsf{else}\ y$ should have the tuple type $\mathsf{qbit} \otimes \mathsf{qbit}$ rather than just qbit since the control qubit should live after the execution of the term.

A naïve generalization of this construct to possibly higher-order N_1 and N_2 would have the following typing rule:

$$\frac{\Gamma \vdash M : \mathsf{qbit} \qquad \Gamma' \vdash N_1 : A \qquad \Gamma' \vdash N_2 : A}{\Gamma, \Gamma' \vdash \mathsf{qif}\, M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 : \mathsf{qbit} \otimes A} \tag{$*$}$$

This naïve typing rule for quantum control allows some physically unrealizable program to be typed, when A is higher-order.

The following program reveals a critical issue:

$$\mathsf{let}\, x' \otimes f : \mathsf{qbit} \otimes (\mathsf{qbit} \multimap \mathsf{qbit}) = \big(\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\ \mathsf{else}\ \mathrm{id}\big)\ \mathsf{in}\ f(x') : \mathsf{qbit}. \tag{$**$}$$

In this program, the qif term first generates a X gate controlled by a qubit x, that is CX gate. It assigns the target X gate to f, and renames the control qubit to x'. Then, it applies f to x' itself. This implies that the qubit x used for control must again be used as the target qubit of the CX gate. As illustrated in Figure 1, this results in a closed timelike curve, violating causality.

It is also possible to explain the problem of the program (∗∗) without appealing to diagrammatic intuition. Let **Hilb** be the category of finite-dimensional Hilbert spaces and all linear maps. Since **Hilb** is a compact closed category and has the coproduct $\mathsf{qbit} = 1 + 1$, the above program can be interpreted in **Hilb** in the standard way. The resulting denotation is a linear map $\mathbb{C}^2 \longrightarrow \mathbb{C}^2$. We give an explicit description of this linear map. For $x = |0\rangle$, we have $x' = |0\rangle$ and $f = \mathrm{id}$, so the outcome is $f(x') = |0\rangle$. Similarly, for $x = |1\rangle$, we have

$x' = |1\rangle$ and $f = \mathsf{X}$, so the outcome is $f(x') = |0\rangle$. So the denotation of the above program is $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$. However, since physically realizable linear maps of type $\mathsf{qbit} \multimap \mathsf{qbit}$ are only unitaries, the above program is not physically implementable.

Simmons and Kissinger's Causal Logic. The above discussion shows that, when the type of the control target A is a function type, allowing interaction between the control qubit and the controlled function leads to a problem. Hence, we need a mechanism that properly regulates such interaction.

To clarify what kind of mechanism would be required, let us examine how quantum conditionals controlling functions could be implemented. A point is that, although it is unclear how to implement $\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\ \mathsf{else}\ \mathrm{id}$, it is clear what is $\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\, L\ \mathsf{else}\ \mathrm{id}\, L$ for given L, which is the controlled-X gate applied to (x, L). Our idea is that a quantum conditional branching that controls functions should not be executed immediately; instead, its execution should be deferred until the argument of the controlled function is determined. Guided by this intuition, we design a type system that avoids the above problem.

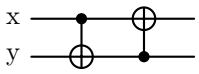
The above discussion suggests that the type of $\mathsf{qif}\, M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2$ should not be $\mathsf{qbit} \otimes A$, but rather a type expressing that "qbit becomes available only after A." Fortunately, a logical connective with precisely this intended meaning have already been studied, namely the *seq-connective* $\lhd$ from pomset logic [47] and BV-logic [25]. These logics are both based on multiplicative linear logic (MLL) with MIX rule, conservatively extended by adding $A \lhd B$ whose intuitive meaning is the causal relation "A happens *before* B".

By adopting this idea, we can now fix the typing rule $(*)$ to the following by restricting the usage of the control qubit to only after A is "resolved" (where $A \rhd B := B \lhd A$):

$$\frac{\Gamma \vdash M \colon \mathsf{qbit} \qquad \Gamma' \vdash N_1 \colon A \qquad \Gamma' \vdash N_2 \colon A}{\Gamma, \Gamma' \vdash \mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2 \colon \mathsf{qbit} \rhd A}\ .$$

We shall prove that this new typing rule actually solves the problem.

However, this mechanism is too restrictive on its own. For example, the following circuit



is obviously physically realizable, but the corresponding program

$$\mathsf{let}\,(x', y') = \big(\mathsf{qif}\, x\ \mathsf{then}\ \mathsf{X}\, y\ \mathsf{else}\ y\big)\ \mathsf{in}\ \big(\mathsf{qif}\, y'\ \mathsf{then}\ \mathsf{X}\, x'\ \mathsf{else}\ x'\big)$$

is not well-typed, since for $(x', y') \colon \mathsf{qbit} \rhd \mathsf{qbit}$, the first qbit x' will be only available after the second qbit y' has been consumed.

Here, a concept of *first-order proposition* introduced by Simmons and Kissinger [52] comes into play. The idea is to equip a "causally simple" proposition with the attribute "first-order" and to admit the following additional rules for first-order propositions (where F is first order and A is arbitrary):

$$F \otimes A \iff F \lhd A \qquad F \parr A \iff F \rhd A.$$

Simmons and Kissinger's *causal logic* [52] is pomset logic [47] enriched with the notion of first-order propositions. In causal logic, by considering qbit as a first order type, we have the type isomorphism $\mathsf{qbit} \otimes \mathsf{qbit} \iff \mathsf{qbit} \rhd \mathsf{qbit}$, which solves our concern.

Based on this intuition, this paper proposes a higher-order quantum programming language with quantum conditional branching, equipped with a type system grounded in causal logic. For technical reasons, our type system is based on a weaker version of causal logic, namely *intuitionistic BV logic* [3] extended with the notion of first-order propositions. We show that the type system behaves as intended: every program of type $\mathsf{qbit} \multimap \mathsf{qbit}$ denotes a unitary transformation (Proposition 39).

Our type system is important not only because it resolves the aforementioned issue concerning causality, but also because it provides new evidence for the relevance of causality-aware logics to higher-order quantum computation. Previous work showed that structures of causality-aware logics can be found in causal models motivated by quantum computation. This is undoubtedly an important observation. However, it was not clear whether these structures appeared merely incidentally, or whether they are essential to higher-order quantum computation. This paper demonstrates that causality-aware logics can play an active role in solving a concrete problem, thereby providing new evidence for their relevance.

Categorical Model. We also construct a new categorical model. This model is used to prove the soundness of our type system, and it also answers the previously-mentioned question of whether one can build a model consisting solely of pure supermaps.

Our construction is motivated by the *Caus construction* [35, 51, 52]. It is a categorical construction that produces a model of causal logic (hence models BV-logic and pomset-logic) from a compact closed category with some additional structure, called discarding maps $\overset{\cdot}{\top}_A \colon A \longrightarrow I$. A particular example of such category is **CPM**, which is the category of completely positive maps. By the Caus construction, we can construct the category **Caus**[**CPM**] that can capture the causal structure of supermaps.

However, this category is insufficient for two reasons: (i) it does not rule out non-pure supermaps, such as the OCB process; and (ii) it does not interpret quantum conditionals. In particular, the latter is critical. Let us consider the program $\mathsf{qif}\ x\ \mathsf{then}\ U\, y\ \mathsf{else}\ y$, whose semantics should be the controlled unitary CU. To define a compositional semantics, we should be able to construct CU from U, but the map $U \mapsto CU = (|1\rangle\langle 1| \otimes U) + (|0\rangle\langle 0| \otimes \mathrm{id})$ turns out be not completely positive because it respects global phases.

Our idea is to leverage the connection between **Hilb** and **CPM**: we define **CausHilb** as the pullback of **Caus**[**CPM**] $\longrightarrow$ **CPM** via **Hilb** $\longrightarrow$ **CPM**. A high-level categorical argument establishes that **CausHilb** carries the expected logical structure.

Related Work. Some closely related work has already been discussed above.

The quantum SWITCH was first discovered in [17], and later, many different kinds of supermaps with indefinite causal order have been found, including the OCB process [41], the Lugano process [9, 10], and the Grenoble process [55].

The class of supermaps that are smaller than pure supermaps includes extensibly causal maps [42, 22] which are maps that do not violate causal inequality [1, 13, 10], and QC-QC [55] which is defined as a specific procedure to create a superposition of causal order. The QC-QC is included in the class of extensibly causal maps, but equality of these two classes remains an open problem. Our approach is also different from these studies in that we can describe classes of higher-order functions. We also conjecture that the supermaps expressible in the language we present define a subclass of these two classes.

Pomset-logic was introduced in 1997 [47, 48] as a logic whose correctness is defined by proof-nets. On the other hand, BV-logic was introduced in the series of papers "A system of interaction and structure" [25, 54, 40, 14, 26]. The first paper [25] introduced BV-logic

in 2007, and the second paper [54] discussed the necessity of deep inference. The third paper was initially intended to prove the equivalence of pomset-logic and BV-logic. However, although this had long been believed, it was disproved considerably later, and this negative result for the conjecture became the third paper [40] in 2023. The fourth [14] and fifth [26] papers concern exponentials. An intuitionistic variant of BV-logic was published recently in [3]. There is prior work relating these logics to the π-calculus [2, 31], but, to the best of our knowledge, this is the first paper to incorporate BV-logic into a programming language via the Curry–Howard correspondence in a setting not based on the π-calculus.

Apart from the Caus construction, there are some categorical studies of supermaps. In [27], Heffords and Wilson have studied semantics of supermaps based on strong profunctors, and in [28], they also studied the Chu construction [8], which is known as a general construction that creates a $*$-autonomous category from a monoidal closed category, and yields a BV-category from a duoidal category. Other than that, Li and Zamdzhiev [38] have studied the BV-structure in the category of operator spaces.

There is also a study that relates causal relations in physics with linear logic in [30, 33]. These work can be compared with causal logic [52]. These studies appear to be based on similar concepts, yet the structures of union and intersection seem to differ.

Outline. We first review some preliminaries in Section 2, and review BV logic in Section 3. In Section 4, we introduce our language λ^{QIF} together with a simple (degenerate) denotational semantics in **Hilb**. In Section 5, we define our categorical model of pure supermaps and show that it is a BV-category. Finally, in Section 6, we study the semantics of λ^{QIF} in detail: we prove a full-abstraction theorem and investigate term definability.

2 Preliminaries

2.1 Linear Logic: Proof Systems and Models

In this section, we briefly review the terminology and background on linear logic that will be used throughout the paper. For further details, we refer the reader to the standard literature.

Linear logic, introduced by Girard [24], is a logic in which the structural rules of weakening and contraction are not freely available. This restriction corresponds to disallowing discarding and copying, and for this reason linear logic is often described as a *resource-aware* logic.

Linear logic has several well-known fragments. One of them is multiplicative intuitionistic linear logic (MILL), which features the connectives $\otimes$, $\multimap$ and the unit I. One may view $\otimes$ and $\multimap$ as linear analogues of pairing and functions, and compare them with conjunction $\wedge$ and implication $\to$ in ordinary logic. A crucial difference, however, is that $\otimes$ is not idempotent, *i.e.*, A and $A \otimes A$ represent different amounts of resources. In particular, $A \otimes B \vdash A$ is not derivable, since B cannot be discarded. The Curry–Howard–Lambek correspondence between MILL, the linear lambda calculus, and symmetric monoidal closed categories is well known.

A larger fragment containing MILL is multiplicative linear logic (MLL). It can be seen as extending MILL with negation $A^{\perp}$, which allows one to define $\parr$, the De Morgan dual of $\otimes$, as $A \parr B := (A^{\perp} \otimes B^{\perp})^{\perp}$. The categorical structure corresponding to MLL is a *$*$-autonomous category*, which is defined as follows.

▶ **Definition 1.** *A $*$-autonomous category is a symmetric monoidal closed category $(\mathcal{C}, \otimes, I, \multimap)$ equipped with a fully faithful functor $(-)^* \colon \mathcal{C}^{\mathrm{op}} \longrightarrow \mathcal{C}$ and a natural isomorphism $A \multimap B^* \cong (A \otimes B)^*$.* ◀

In particular, among $*$-autonomous categories, there is a somewhat degenerate class known as *compact closed categories*. These are categories in which the two monoidal structures $\otimes$ and $\parr$ coincide. They can be defined as a monoidal category with *unit* $I \longrightarrow A^* \otimes A$ and *counit* $A \otimes A^* \longrightarrow I$ satisfying some axioms.

It is also common to extend MLL with the rule $A \otimes B \vdash A \parr B$. This rule is called the *MIX* rule. Categorically, it corresponds to equipping a $*$-autonomous category with a morphism $\bot \to I$ satisfying the appropriate coherence conditions. In particular, when this morphism is an isomorphism, such a category is called an *isoMIX category* [19].

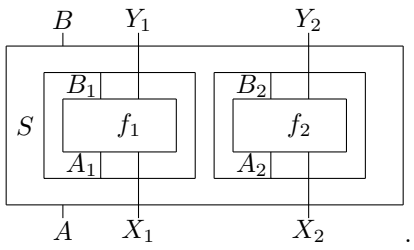
2.2 Basics of Quantum Computation and Supermaps

In this section, we briefly review basic notions from quantum computation. For precise definitions, see, for example, [50, 49].

A (pure) quantum state is represented by a norm-1 element of a Hilbert space. In this paper we restrict attention to the finite-dimensional setting, so a pure state is a vector $v \in \mathbb{C}^n$ with $\|v\| = 1$. We write the standard basis vectors $e_0, \dots, e_{n-1} \in \mathbb{C}^n$ as $|0\rangle, \dots, |n-1\rangle$. For qubits, the basis states $|0\rangle$ and $|1\rangle$ are often identified with the Booleans `false` and `true`. Unlike the classical case, a qubit state need not be $|0\rangle$ or $|1\rangle$; it can also be any linear combination $\alpha|0\rangle + \beta|1\rangle$, called a *superposition* state. In quantum computation one also considers *mixed states*, which represent probabilistic mixtures of pure states. Mathematically, a mixed state on $\mathbb{C}^n$ is a *density matrix*, *i.e.*, a positive semidefinite operator $\rho \in \mathbb{C}^{n\times n}$ with $\mathrm{Tr}(\rho) = 1$. Equivalently, ρ can be written as a convex combination of rank-one projectors: $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$ where $p_i \geq 0$, $\sum_i p_i = 1$, $\|\psi_i\| = 1$. A pure state $|\psi\rangle$ corresponds to the special case $\rho = |\psi\rangle\langle\psi|$, which has rank 1.

To describe general state transformations, we use *quantum channels*. Formally, a quantum channel is a completely positive trace-preserving (CPTP) linear map $\Phi : \mathcal{L}(\mathbb{C}^n) \to \mathcal{L}(\mathbb{C}^m)$, where $\mathcal{L}(\mathbb{C}^n)$ denotes the set of $n \times n$ matrices. A particularly important subclass consists of channels that preserve purity. Such a channel is implemented by an *isometry* $V : \mathbb{C}^n \to \mathbb{C}^m$ satisfying $V^\dagger V = I$, via $\Phi(\rho) = V\rho V^\dagger$. We refer to these as *pure channels*. Every channels can be decomposed to pure channels followed by some qubit discardings.

We also consider *higher-order* transformations that take quantum channels as inputs. In particular, it should preserve the trace-preserving property. Moreover, when represented in a matrix form (e.g. via a Choi–Jamiołkowski-type representation), the supermap itself is *completely positive*. Following the definition in [56], we define *pure supermaps* as a linear map $S\colon \left(\bigotimes_{n=1}^{N} \mathcal{L}(A_i) \multimap \mathcal{L}(B_i)\right) \longrightarrow \mathcal{L}(A) \multimap \mathcal{L}(B)$ such that for all auxiliary (finite-dimensional) Hilbert spaces X_i and Y_i and isometry operators $f_i\colon A_i \otimes X_i \multimap B_i \otimes Y_i$, the partial application to S described as follows defines an isometry $A \otimes_i X_i \multimap B \otimes_i Y_i$:



2.3 Definition of Basic Categories

We define two basic categories **Hilb** and **CPM** which are both standard model of quantum computation. Both categories are compact closed.

▶ **Definition 2.** *The category* **Hilb** *consists of the set of objects* $\mathrm{Ob}(\mathbf{Hilb}) = \mathbb{N}$ *where the maps* $\mathbf{Hilb}(n, m)$ *is defined by the linear maps* $\mathbb{C}^n \longrightarrow \mathbb{C}^m$.

This category **Hilb** *is compact closed where the structures are defined as follows:*

- *The monoidal unit is* 1*, and the monoidal product of objects* $n \otimes m$ *is* $n \times m$.
- *The monoidal product of maps are defined by the tensor product,* i.e.*, for linear maps* $f \in \mathbf{Hilb}(n, n')$ *and* $g \in \mathbf{Hilb}(m, m')$*, when* $f\,|i\rangle = \sum_{i'} \alpha_{i'}\,|i'\rangle$ *and* $g\,|j\rangle = \sum_{j'} \beta_{j'}\,|j'\rangle$*, let* $(f \otimes g)(|i\rangle \otimes |j\rangle) := \sum_{i',j'} \alpha_{i'}\beta_{j'}(|i'\rangle \otimes |j'\rangle)$.
- *The monoidal structure is strict: the associater* $\alpha\colon (A \otimes B) \otimes C \longrightarrow A \otimes (B \otimes C)$ *and the uniters* $\ell\colon I \otimes A \longrightarrow A$ *and* $r\colon A \otimes I \longrightarrow A$ *are identity.*
- *The dual of an object is itself* $n^* = n$*, and the dual map* $f^*\colon m \longrightarrow n$ *of a map* $f\colon n \longrightarrow m$ *is defined by the transpose.*
- *The unit map* $\eta\colon 1 \longrightarrow n \otimes n$ *and the counit map* $\varepsilon\colon n \otimes n \longrightarrow 1$ *are defined as follows:*

$$\eta\colon 1 \longmapsto \textstyle\sum_{i=0}^{n} |i\rangle \otimes |i\rangle\,, \qquad \varepsilon\colon |i\rangle \otimes |j\rangle \longmapsto \delta_{ij}.$$

It also admits finite biproducts defined by $n \oplus_{\mathbf{Hilb}} m := n + m$. ◀

▶ **Definition 3.** *The category* **CPM** *has sequences of natural numbers* $\vec{n} = (n_1, \ldots, n_k)$ *as objects, and each map* $f \in \mathbf{CPM}(\vec{n}, \vec{m})$ *is defined by a matrix of maps* $(f_{ij})\colon \vec{n} \longrightarrow \vec{m}$ *where* $f_{ij}\colon \mathcal{L}(\mathbb{C}^{n_i}) \longrightarrow \mathcal{L}(\mathbb{C}^{m_j})$ *is completely positive. The composition is defined by the matrix multiplication,* i.e.*,* $(g_{k\ell}) \circ (f_{ij}) = (\sum_i g_{j\ell} \circ f_{ij})$.

This category is also compact closed with the following structures:

- *The monoidal unit is* (1)*, and the monoidal product* $\vec{n} \otimes \vec{m}$ *is given by*

$$(n_1m_1,\, n_1m_2,\, \ldots,\, n_1m_\ell,\, n_2m_1,\, \ldots,\, n_km_\ell).$$

- *The monoidal product of morphisms* $(f_{ij}) \otimes (g_{i'j'})$ *is defined by the pointwise Kronecker product of CPMs* $(f_{ij} \otimes g_{i'j'})$.
- *It is strict monoidal.*
- *The dual of an object is itself* $\vec{n}^* = \vec{n}$*, and the dual of* (f_{ij}) *is defined by* (f^*_{ji}) *where* f^*_{ji} *is the transpose of* f_{ij}.
- *The unit and counit are defined by* $(\eta_{1,ij})$ *and* $(\varepsilon_{ij,1})$ *where*

$$\eta_{1,ij} = \begin{cases} 1 \longmapsto \sum_{k,\ell \in \{0,\ldots,n_i-1\}} |k\rangle\langle\ell| \otimes |k\rangle\langle\ell| & \text{if } i = j \\ 1 \longmapsto 0 & \text{otherwise,} \end{cases}$$

$$\varepsilon_{ij,1} = \begin{cases} |k\rangle\langle\ell| \otimes |k'\rangle\langle\ell'| \longmapsto \delta_{k,k'}\delta_{\ell\ell'} & \text{if } i = j \\ |k\rangle\langle\ell| \otimes |k'\rangle\langle\ell'| \longmapsto 0 & \text{otherwise.} \end{cases}$$

It also admits finite biproducts defined by the concatination of sequences $\vec{n} \oplus_{\mathbf{CPM}} \vec{m} = \vec{n}\vec{m}$. ◀

These two categories are related with the following *embedding functor.*[2]

▶ **Definition 4.** *There is a functor* $\iota\colon \mathbf{Hilb} \longrightarrow \mathbf{CPM}$ *that sends* n *to* (n) *and* f *to the map* $\lambda\rho.\ f \circ \rho \circ f^\dagger$*. This functor preserves all the compact closed strucuture* on the nose*, but it does not preserve biproducts.* ◀

[2] We call this functor embedding since it embeds pure quantum computation into general quantum computation even though it is not full or faithful.

Since these two categories are compact closed, they can model a linear lambda calculus. The category **Hilb** can interpret pure quantum computation: the semantics of the qubit type can be defined by $2 = 1 \oplus_{\mathbf{Hilb}} 1$, where a state of a qubit $I = 1 \longrightarrow 2$ corresponds with a vector, and the semantics of a program can be given as a linear functions. On the other hand, **CPM** can interpret any quantum computation including measuremets or discardings. Through the embedding functor ι, the pure quantum computation can also be interpreted in **CPM**. Additionally, the semantics of Booleans can be defined by $(1,1) = 1 \oplus_{\mathbf{CPM}} 1$, and the measurement and discarding maps are defined as follows:

$$\begin{array}{rcccc} m := (P_i)_{i=1,2}\colon & (2) & \longrightarrow & (1,1) \\ \text{where } P_i\colon & M & \longmapsto & M_{ii}, \end{array} \qquad \begin{array}{rcccc} \dot{\top}_n := (\mathrm{Tr}_{n_i})_{i=1,\dots,k}\colon & \vec{n} & \longrightarrow & (1) \\ \text{where } \mathrm{Tr}_m\colon & M & \longmapsto & \sum_{j=1}^{m} M_{jj}. \end{array}$$

3 BV-logic

In addition to linear logic, BV-logic [25] and pomset-logic [47] introduce a *non-commutative* and *self-dual* connective $\lhd$ called *seq* or *before* to the logic. That is, for each pair of propositions A and B, unlike other connectives such as $\otimes$, $\parr$, & or $\oplus$, there is no symmetry rule $A \lhd B \implies B \lhd A$, and it admits De Morgan duality with itself $(A \lhd B)^\perp \iff A^\perp \lhd B^\perp$. Both BV-logic and pomset-logic are conservative extensions of MLL + nullary and binary MIX.

BV-logic can be seen as a linear logic with causality relation, where $A \lhd B$ means A happens before B. We list below two illustrative derivations in BV.

$$A \otimes B \vdash A \lhd B \vdash A \parr B \qquad (A \lhd C) \otimes (B \lhd D) \vdash (A \otimes B) \lhd (C \otimes D)$$

In linear logic, $A \otimes B$ can be understood as a pair of resources A and B. Here, in BV-logic, we can additionally assume $A \otimes B$ as a pair *without causal ordering*. Therefore, the first derivation above $A \otimes B \vdash A \lhd B$ can be interpreted as "from a pair of resources $A \otimes B$, we can force a causal relation to obtain $A \lhd B$". Dually, $A \parr B$ can be understood as a pair *with some causation*. So $A \lhd B \vdash A \parr B$ can be interpreted as "by forgetting the detail of causal relation $A \lhd B$, we obtain a pair of resource with unknown causation $A \parr B$". The second is the *interchange rule*, which can also be understood similarly as, "by assuming extra causal relations from A to D and from B to C, we can obtain $(A \otimes B) \lhd (C \otimes D)$ from $A \lhd C$ and $B \lhd D$". Similar interchange rules also appear in other logic literature that addresses causality in a context of concurrency [29, 44].

BV-logic is famous for its style of inference rules called *deep inference* [54]. Deep inference can be seen as a special kind of sequent calculus, and adopted by BV-logic to admit cut elimination. While usual sequent calculi have rewriting rules that only pattern match the outermost connective of a formula, deep inference allows one to pattern match a formula recursively at arbitrary depth, described as follows:

$$\frac{A \vdash B}{S\{A\} \vdash S\{B\}} \quad \text{where} \quad \begin{array}{l} S\{\cdot\} ::= \{\cdot\} \mid A \otimes S \mid S \otimes A \mid A \lhd S \\ \qquad\qquad \mid S \lhd A \mid A \parr S \mid S \parr A. \end{array}$$

3.1 Intuitionistic BV

Since we use a logic in this paper to define a type system for a lambda calculus, it is natural, via the Curry-Howard correspondence, to consider the intuitionistic variant of BV logic.

Intuitionistic BV-logic (IBV) was introduced in [3]. In IBV, we have $\multimap$ instead of $\parr$ as in MILL. IBV is proven to be a conservative extension of MILL. It is also defined by deep inference rules so that it admits cut elimination.

$$A \vdash A \qquad \frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \qquad \frac{\Gamma \vdash A \qquad \Gamma' \vdash B}{\Gamma, \Gamma' \vdash A \otimes B} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} \qquad \frac{\Gamma \vdash A \qquad \Gamma', B \vdash C}{\Gamma, \Gamma', A \multimap B \vdash C}$$

$$\frac{\Gamma \vdash A}{\Gamma, I \vdash A} \qquad \vdash I \qquad \frac{\Gamma \vdash A \qquad \Gamma', A \vdash B}{\Gamma, \Gamma' \vdash B} \qquad \vdash I \lhd I \qquad \frac{\Gamma, A \vdash B \qquad \Gamma', C \vdash D}{\Gamma, \Gamma', A \lhd C \vdash B \lhd D}$$

$$\frac{\Gamma \vdash A \lhd C \qquad \Gamma' \vdash B \lhd D}{\Gamma, \Gamma' \vdash (A \otimes B) \lhd (C \otimes D)} \qquad \frac{\Gamma \vdash (A \multimap C) \lhd (B \multimap D)}{\Gamma \vdash (A \lhd B) \multimap (C \lhd D)} \qquad (A \lhd B) \lhd C \dashv\vdash A \lhd (B \lhd C)$$

Figure 2 Derivation rules in sequentIBV.

We present a simplified sequent calculus *sequentIBV* for IBV in Figure 2. Instead of having deep inference rules to admit cut-elimination, we rather define it as a usual sequent calculus. This also allows us to significantly reduce the number of rules which were needed in IBV to comprehensively represent all normal forms. As we will see in the next section, the correspondence between proofs and programs becomes much clearer in this simpler sequent calculus style definition.

As expected, sequentIBV and IBV have the same expressivity.

▶ **Proposition 5.** *$A \vdash B$ is derivable in sequentIBV if and only if $A \multimap B$ is derivable in IBV [3].*

Sketch of Proof. Checking that all sequents in IBV are provable in sequentIBV is easy. One can also check that all basic derivations in IBV can be also proven in sequentIBV. To check that all derivations in IBV that use the deep inference principle in the proof are also derivable in sequentIBV, it suffices to prove that, in sequentIBV, if $X \vdash Y$ is derivable, then $S\{X\} \vdash S\{Y\}$ (or $S\{Y\} \vdash S\{X\}$ if the hole in S is at the negative position) is also derivable for each following S defined by

$$S\{\cdot\} ::= \{\cdot\} \mid A \otimes S \mid S \otimes A \mid A \lhd S \mid S \lhd A \mid A \multimap S \mid S \multimap A.$$

For example, if $S = \{\cdot\} \lhd A$, this can be done as follows:

$$\frac{\dfrac{\pi}{X \vdash Y} \qquad A \vdash A}{X \lhd A \vdash Y \lhd A}.$$

◀

3.2 First Order Proposition in Causal Logic

Simmons and Kissinger [52] defined *causal logic* as an extension of pomset-logic obtained by adding *first-order propositions*. This is a particularly natural extension: for example, in the proof-net style proof of linear logic, it is shown that, in an appropriate sense, adding the seq-connective is equivalent to adding first-order propositions [52, Proposition 4.13]. Concretely, these first-order propositions satisfy the following.

$$\forall A\text{: first-order}, \quad A \otimes B \dashv\vdash A \lhd B \quad \text{and} \quad B \lhd A \dashv\vdash B \parr A$$

Recalling that $A \otimes B \vdash A \lhd B \vdash A \parr B$ is already derivable in BV or pomset-logic, this means we can prove the converse if we assume first-orderness for some propositions. Such first-order propositions can be seen as "data without input" or "data without past". So, when A is

$$
\begin{array}{rrcl}
\textit{Types} & A, B & ::= & d\ (\in \mathbb{N}^+) \mid \bot \mid A \otimes B \mid A \lhd B \mid A \multimap B \\
\textit{First-order types} & F : \mathrm{FO} & ::= & d \mid \bot \mid F \otimes F \mid F \lhd F \\
\textit{Terms} & M, N & ::= & x \mid () \mid \lambda x.\, M \mid M\, N \mid M \otimes N \\
 & & \mid & \mathsf{let}\ \_ = M\ \mathsf{in}\ N \mid \mathsf{let}\, x \otimes y = M\ \mathsf{in}\ N \\
 & & \mid & M \lhd N \mid \mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2) \\
 & & \mid & \mathsf{assoc}_L^{\lhd}(M) \mid \mathsf{assoc}_R^{\lhd}(M) \mid \mathsf{interch}(M) \\
 & & \mid & \mathsf{await}(M) \mid \mathsf{expose}(M) \\
 & & \mid & \lceil f \rceil \mid |0\rangle \mid U \mid \mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2
\end{array}
$$

where f in $\lceil f \rceil$ is a type isomorphism, defined at (1).

Abbreviations: $\mathsf{qbit} := 2$, $(\mathsf{let}\, x = M\ \mathsf{in}\ N) := (\lambda x.\, N)M$, $A \rhd B := B \lhd A$.

Figure 3 The syntax of λ^{QIF}.

first-order, the causal relation $A \lhd B$ says "B can be obtained after A is obtained", but since there is nothing to wait to obtain A, this is simply the same as having a pair $A \otimes B$.

In particular, in our intuitionistic setting, we can replace $B \lhd A \dashv\vdash B \parr A$ with $B \multimap (C \lhd A) \dashv\vdash (B \multimap C) \lhd A$, where the left-to-right implication is always possible.

4 Language for Pure Quantum Computation

We now present our language λ^{QIF}. Our language is *pure* (measurement-free), *higher-order*, and has *quantum conditional branching*. We also have a type system inspired by IBV, so we have the $\lhd$ type representing causal ordering.

4.1 Syntax

We define the syntax of our language λ^{QIF} in Figure 3.

Types. For the types, we have d (a positive natural number) that represents qu*d*its, which are quantum data types of d dimensions. Our language is based on Intuitionistic BV-logic, and has $\otimes$ (tuple type), $\bot$ (unit type), $\lhd$ (causal ordering), and $\multimap$ (function type). We have first-order types, corresponding to first-order propositions. The atomic types d and $\bot$ are first-order, and they are closed under $\otimes$ and $\lhd$, but not under $\multimap$.

Terms. In our language λ^{QIF}, we have terms that are derived from linear lambda calculus, BV-logic, and pure quantum computation.

In the first two lines of Figure 3, we have some terms taken from standard linear lambda calculus. We have x (variable), $()$ (the value of the type $\bot$), $\lambda x.M$ (lambda abstraction), $M\,N$ (function application), $M \otimes N$ (pair of terms), and two let bindings corresponding to $\bot$ and $\otimes$. The usual let binding can be derived as $(\lambda x.N)M$.

In the next two lines, we have some terms derived from BV-logic. We have $M \lhd N$ (causally ordered terms) and a let binding for $\lhd$. We also have explicit associators $\mathsf{assoc}_R^{\lhd}$ and $\mathsf{assoc}_L^{\lhd}$ for $\lhd$, which are the inverse of each other. The term $\mathsf{interch}$ corresponds to the interchange rule in BV-logic.

$$x\colon A \vdash x\colon A \qquad \vdash ()\colon \bot \qquad \frac{\Gamma, x\colon A \vdash M\colon B}{\Gamma \vdash \lambda x.M\colon A \multimap B} \qquad \frac{\Gamma \vdash M\colon A \multimap B \qquad \Gamma' \vdash N\colon A}{\Gamma, \Gamma' \vdash M\,N\colon B}$$

$$\frac{\Gamma \vdash M\colon A \qquad \Gamma' \vdash N\colon B}{\Gamma, \Gamma' \vdash M \otimes N\colon A \otimes B} \qquad \frac{\Gamma \vdash M\colon \bot \qquad \Gamma' \vdash N\colon A}{\Gamma, \Gamma' \vdash \mathsf{let}\ \_ = M\ \mathsf{in}\ N\colon A}$$

$$\frac{\Gamma \vdash M\colon A \otimes B \qquad x\colon A, y\colon B, \Gamma' \vdash N\colon C}{\Gamma, \Gamma' \vdash \mathsf{let}\, x \otimes y = M\ \mathsf{in}\ N\colon C} \qquad \frac{\Gamma \vdash M\colon (A \lhd B) \lhd C}{\Gamma \vdash \mathsf{assoc}^{\lhd}_{L}(M)\colon A \lhd (B \lhd C)}$$

$$\frac{\Gamma \vdash M\colon A \lhd (B \lhd C)}{\Gamma \vdash \mathsf{assoc}^{\lhd}_{R}(M)\colon (A \lhd B) \lhd C}$$

SEQ-INTRO

$$\frac{\Gamma \vdash M\colon A \qquad \Gamma' \vdash N\colon B}{\Gamma, \Gamma' \vdash M \lhd N\colon A \lhd B}$$

LET-SEQ

$$\frac{\Gamma_0 \vdash M\colon A \lhd B \qquad x\colon A, \Gamma_1 \vdash N_1\colon C \qquad y\colon B, \Gamma_2 \vdash N_2\colon D}{\Gamma_0, \Gamma_1, \Gamma_2 \vdash \mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2)\colon C \lhd D}$$

INTERCH

$$\frac{\Gamma \vdash M\colon (A \lhd C) \otimes (B \lhd D)}{\Gamma \vdash \mathsf{interch}(M)\colon (A \otimes B) \lhd (C \otimes D)}$$

AWAIT

$$\frac{\Gamma \vdash M\colon A \lhd B \qquad A\colon \mathrm{FO}}{\Gamma \vdash \mathsf{await}(M)\colon A \otimes B}$$

EXPOSE

$$\frac{\Gamma \vdash M\colon B \multimap (C \lhd A) \qquad A\colon \mathrm{FO}}{\Gamma \vdash \mathsf{expose}(M)\colon (B \multimap C) \lhd A}$$

TYPE-ISO

$$\frac{f\colon A \cong B}{\vdash \lceil f \rceil\colon A \multimap B}$$

QINIT

$$\vdash |0\rangle : d$$

UNITARY

$$\frac{U\colon \mathbb{C}^n \longrightarrow \mathbb{C}^n; \mathrm{Unitary}}{\vdash U\colon n \multimap n}$$

QIF

$$\frac{\Gamma \vdash M\colon \mathsf{qbit} \qquad \Gamma' \vdash N_1\colon A \qquad \Gamma' \vdash N_2\colon A}{\Gamma, \Gamma' \vdash \mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2\colon \mathsf{qbit} \rhd A}$$

Figure 4 Typing rule of λ^{QIF}.

The terms await and expose are the special rules for first-order types, which will be explained along with the typing rules in Section 4.2.

In the last line, we have some quantum primitives. We have $|0\rangle$ (qubit initialization), U (unitary map), and qif (quantum conditional). We also have a casting $\lceil f \rceil$ along a type isomorphism $f\colon A \cong B$, where type isomorphism is defined as a congruence relation as

$$1 \cong \bot, \quad n \otimes m \cong n \times m. \tag{1}$$

4.2 Typing Rules

The typing rules for λ^{QIF} are given in Figure 4. Typing rules for terms derived from linear lambda calculus are standard and can be found in the literature.

We explain the terms that are derived from BV-logic. The two associators have the type as expected. With the typing rule SEQ-INTRO, we can introduce a term of $\lhd$-type. This rule, for example, allows us to derive a term of type $A \otimes B \multimap A \lhd B$ as

$$\vdash \lambda x.\, \mathsf{let}\, y \otimes z = x\ \mathsf{in}\ (y \lhd z)\colon A \otimes B \multimap A \lhd B.$$

With LET-SEQ, we can destruct a term of type $A \lhd B$. This corresponds to the functoriality of $\lhd$. The rule INTERCH corresponds to the interchange law in BV-logic. Notably, from this rule, we can also derive its dual, as in Figure 5.

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash M\colon (A \multimap C) \lhd (B \multimap D)}{\Gamma, x\colon A \lhd B \vdash M \otimes x\colon ((A \multimap C) \lhd (B \multimap D)) \otimes (A \lhd B)} \qquad y\colon (A \multimap C) \otimes A \vdash \mathsf{let} \cdots : C}{\Gamma, x\colon A \lhd B \vdash \mathsf{interch}(M \otimes x)\colon ((A \multimap C) \otimes A) \lhd ((B \multimap D) \otimes B)} \qquad z\colon (A \multimap C) \otimes A \vdash \mathsf{let} \cdots : D}{\Gamma \vdash \lambda x.\, \mathsf{let}\ y \lhd z = (\mathsf{interch}\, M \otimes x)\ \mathsf{in}\ (\mathsf{let}\ y_1 \otimes y_2 = y\ \mathsf{in}\ y_1\ y_2) \lhd (\mathsf{let}\ z_1 \otimes z_2 = z\ \mathsf{in}\ z_1\ z_2)\colon (A \lhd B) \multimap (C \lhd D)}$$

Figure 5 A term corresponding to the dual of the interchange rule

The AWAIT and EXPOSE rules are the rules that are not derived from the original BV-logic, but from the extension of adding first-order propositions. Intuitively, with the term await, if A is first-order, we can obtain a pair of data A and B from causally ordered data $A \lhd B$ by "waiting" for the computation that produces A to finish. On the other hand, with the term expose, again if A is first-order, we can "expose" the output data A from a function $B \multimap (C \lhd A)$ before it is actually executed, but with a restriction that says A has to be used after the function $B \multimap C$ is resolved. Note that EXPOSE is the only structural term with no corresponding one when we replace every occurrence of $\lhd$ with $\otimes$.

In the remaining terms, we have TYPE-ISO for type casting, QINIT for qu*d*it initialization, and UNITARY for unitary maps. The last rule QIF for quantum conditional statement is notable. The qif term takes a qbit as an input for the condition, and returns data of the type $\mathsf{qbit} \rhd A$, creating the superposition of N_1 and N_2 of type A. An intuition behind this typing rule is that, since the control qubit is required in the execution of A, we have to wait until A is resolved to reuse the control qubit. In fact, for example, when A is given by $B_1 \multimap B_2 \multimap \cdots B_n \multimap n$, the return type $\mathsf{qbit} \rhd A$ has the following equivalence of types

$$\begin{aligned} \mathsf{qbit} \rhd A \quad &\equiv \quad \mathsf{qbit} \rhd (B_1 \multimap B_2 \multimap \cdots B_n \multimap n) \quad \equiv \quad B_1 \multimap (\mathsf{qbit} \rhd B_2 \multimap \cdots B_n \multimap n) \\ \cdots \quad &\equiv \quad B_1 \multimap B_2 \multimap \cdots B_n \multimap (\mathsf{qbit} \rhd n) \quad \equiv \quad B_1 \multimap B_2 \multimap \cdots B_n \multimap (\mathsf{qbit} \otimes n), \end{aligned}$$

which means that the control qubit is returned only after the execution of A is complete. Note that this equivalence relies heavily on the fact that qbit is first-order.

▶ Remark 6. We defined the type of qif by $\mathsf{qbit} \rhd A$ based on IBV $(\otimes, \multimap, \lhd)$, but we could instead define the type as $\mathsf{qbit} \parr A$ and adopt classical MLL types $(\otimes, \parr)$. Indeed, we can have a similar type equivalence with this definition.

$$\mathsf{qbit} \parr (B_1 \multimap B_2 \multimap \cdots B_n \multimap n) \quad \cong \quad B_1 \multimap B_2 \multimap \cdots B_n \multimap (\mathsf{qbit} \parr n),$$

Using $\mathsf{qbit} \parr A$ instead of $\mathsf{qbit} \rhd A$ can be justified by a law of causal logic, $\mathsf{qbit} \rhd A \cong \mathsf{qbit} \parr A$.

The reason why we use IBV and causal logic is as follows. First, classical MLL involves computational complications arising from its classical nature, whereas the intuitionistic system IBV is easier to relate to programs. Second, a type system based on classical MLL needs an additional isomorphism $\mathsf{qbit} \parr n \cong \mathsf{qbit} \otimes n$, of which justification cannot be found except for causal logic. Third, we also believe our explanation that "the control qubit should live after the execution of the branches" intuitively motivates the introduction of $\lhd$. ◀

The next proposition states that our language covers all derivable proofs in IBV.

▶ **Proposition 7.** *Each sequentIBV derivation has a corresponding term in* λ^{QIF}. ◀

▶ Remark 8. Since sequentIBV does not admit cut elimination, our language based on it does not have a natural operational semantics. The language we propose is intended for writing supermaps, and given that not every supermaps inherently possess a natural operational meaning, we deliberately choose not to attempt to define an operational semantics in this paper. ◀

4.3 Examples

▶ **Example 9.** In our language, we can define the following term *SWITCH* that makes the superposition of the order of two function applications as the quantum SWITCH [17]. We write $g \circ f$ to mean $\lambda y.\, g\,(f\,y)$.

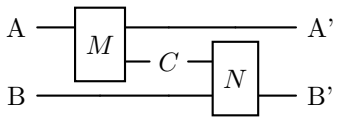
$$x\colon \mathsf{qbit}, f, g\colon A \multimap A \;\vdash\; \mathsf{qif}\; x\; \mathsf{then}\; g \circ f\; \mathsf{else}\; f \circ g \;:\; \mathsf{qbit} \rhd (A \multimap A)$$

When $A = d$ for some $d \in \mathbb{N}^+$, we will later see that the semantics of this term actually defines the quantum SWITCH. In this case, the type equivalence above means that this term is essentially equivalent to the following term

$$x\colon \mathsf{qbit}, f, g\colon d \multimap d \;\vdash\; \lambda y.\, \mathsf{await}(\mathsf{qif}\; x\; \mathsf{then}\; g\,(f\,y)\; \mathsf{else}\; f\,(g\,y)) \;:\; d \multimap (\mathsf{qbit} \otimes d)$$

where the variable y now appears as an input to the term. ◀

▶ **Example 10.** Let us assume that we are given two functions $\Gamma \vdash M\colon A \multimap A' \otimes C$ and $\Gamma' \vdash N\colon C \otimes B \multimap B'$, and we want to connect these via C as in the following circuit.



In our language, we can describe this term with the type $(A \multimap A') \lhd (B \multimap B')$, reflecting the fact that information flows only from $\mathsf{A} = (A \multimap A')$ to $\mathsf{B} = (B \multimap B')$. To do so, we assume that C is first-order, ensuring that no information flows backward through the output C of M. Then we can derive the following terms.

$$\Gamma \vdash \mathsf{expose}(\lambda a.\; \mathsf{let}\, a' \otimes c = M\, a\; \mathsf{in}\; a' \lhd c)\colon (A \multimap A') \lhd C$$
$$\Gamma' \vdash \lambda c.\, \lambda b.\, N\,(c \otimes b)\colon C \multimap (B \multimap B')$$

Let us denote these terms as $\Gamma \vdash M'\colon \mathsf{A} \lhd C$ and $\Gamma' \vdash N'\colon C \multimap \mathsf{B}$. We also have the following term $x\colon C \lhd (C \multimap \mathsf{B}) \vdash L\colon \mathsf{B}$.

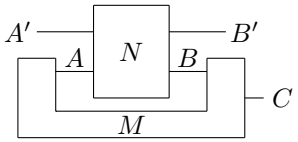
$$x\colon C \lhd (C \multimap \mathsf{B}) \vdash \mathsf{let}\, c \otimes f = \mathsf{await}(x)\; \mathsf{in}\; (f\,c)\colon \mathsf{B}$$

Then we can derive the following

$$\dfrac{\dfrac{\Gamma, \Gamma' \vdash M' \lhd N'\colon (\mathsf{A} \lhd C) \lhd (C \multimap \mathsf{B})}{\Gamma, \Gamma' \vdash \mathsf{assoc}_L^{\lhd}(M' \lhd N')\colon \mathsf{A} \lhd (C \lhd (C \multimap \mathsf{B}))}}{\Gamma, \Gamma' \vdash \mathsf{let}\, a \lhd x = \mathsf{assoc}_L^{\lhd}(M' \lhd N')\; \mathsf{in}\; (a \lhd L)\colon \mathsf{A} \lhd \mathsf{B}}$$

which has the desired type. ◀

▶ **Example 11.** Let us assume we are given a supermap $\Gamma \vdash M\colon (A \multimap B) \multimap C$ and a map $\Gamma' \vdash N\colon A \otimes A' \multimap B \otimes B'$, and we want to connect them via A and B as in the following image to obtain a map of type $A' \multimap B' \otimes C$.

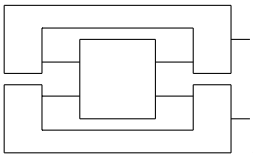


We first assume that B' is first-order. Then we have the following terms M' and N'.

$$\Gamma \vdash M \lhd () \colon ((A \multimap B) \multimap C) \lhd \bot$$
$$\Gamma', a' \colon A' \vdash \mathsf{expose}(\lambda a.\ \mathsf{let}\, b \otimes b' = N(a \otimes a)\ \mathsf{in}\ b \lhd b') \colon (A \multimap B) \lhd B'$$

Now, the term $\mathsf{interch}(M' \otimes N')$ has type $(((A \multimap B) \multimap C) \otimes (A \multimap B)) \lhd (I \otimes B')$. From the left-hand side of $\lhd$, we can obtain C, and from the right-hand side, we can obtain B'. Therefore, there is a term L of type $\Gamma, \Gamma', a' \colon A' \vdash L \colon C \lhd B'$. By assuming that C is also a first-order type, we can obtain the term $\Gamma, \Gamma' \vdash \lambda a'.\ \mathsf{await}(L) \colon A \multimap C \otimes B'$.

In particular, when we are given another supermap of type $(A' \multimap B') \multimap C'$ as an input, where C' is also FO, we can connect A' and B' again to obtain $C \otimes C'$ as in



.

◀

▶ **Example 12.** In our language, we can define the following map

$$x, y \colon \mathsf{qbit}, f, g \colon \mathsf{qbit} \multimap \mathsf{qbit}$$
$$\vdash \mathsf{qif}\ x\ \mathsf{then}\ \mathit{SWITCH}(y, f, g)\ \mathsf{else}\ (f\ y) \rhd g\ :\ \mathsf{qbit} \rhd (\mathsf{qbit} \multimap \mathsf{qbit}),$$

where SWITCH is the term defined in Example 9. The qubit y is used as a control in the `then` branch and as an input of f in the `else` branch. Therefore, this function cannot be represented in a language that separates "control" qubits and "target" qubits as in [18]. ◀

4.4 Degenerate Categorical Semantics

A degenerate categorical semantics for our language can be given in **Hilb**, using the compact closed structure where $n \otimes m = n \lhd m = n \multimap m = n \times m$.

For a type A, its categorical semantics $[\![A]\!]_{\mathbf{Hilb}} \in \mathbf{Hilb}$ is inductively defined as follows:

$$[\![n]\!] = n, \quad [\![\bot]\!] = 1, \quad [\![A \otimes B]\!] = [\![A \lhd B]\!] = [\![A \multimap B]\!] = [\![A]\!] \otimes [\![B]\!].$$

The semantics for terms $\Gamma \vdash M \colon A$ is given in Figure 6 as a morphism $[\![M]\!]_{\mathbf{Hilb}} \colon [\![\Gamma]\!] \to [\![A]\!]$ in **Hilb** where $[\![\Gamma]\!]$ represents the tensor product of the semantics of types in Γ. For most of the terms, the semantics are based on standard linear lambda calculus. For example, the semantics of lambda abstraction and function application are defined via the monoidal closed structure of **Hilb**. As already noted, by identifying $\lhd$ and $\otimes$, all the terms in the first three lines of Figure 6 can be regarded as those in a standard linear lambda calculus. For example, the term $\mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2)$ is identified with $\mathsf{let}\, x \otimes y = M\ \mathsf{in}\ (N_1 \otimes N_2)$, so they are defined to have the same semantics.

In the last line of Figure 6, we have the definition of semantics of await, expose, and the three quantum primitives. For await and expose, the domain and the codomain of the semantics in **Hilb** match, so they can be simply defined as the identities. The semantics of $|0\rangle$ and U are defined by the followings.

$$|0\rangle\ :\ I \longrightarrow d;\ 1 \longmapsto |0\rangle\,, \qquad U\ :\ d \longrightarrow d;\ |\psi\rangle \longmapsto U\,|\psi\rangle\,.$$

$$[\![x\colon A \vdash x\colon A]\!] = \mathrm{id}_{[\![A]\!]} \qquad [\![()]\!] = \mathrm{id}_I \qquad [\![\lambda x.\, M]\!] = \Lambda_{[\![\Gamma]\!],[\![A]\!],[\![B]\!]}[\![M]\!]$$

$$[\![M\ N]\!] = \mathrm{ev}_{[\![A]\!],[\![B]\!]} \circ ([\![M]\!] \otimes [\![N]\!]) \qquad [\![M \otimes N]\!] = [\![M \lhd N]\!] = [\![M]\!] \otimes [\![N]\!]$$

$$[\![\mathsf{let}\ \_ = M\ \mathsf{in}\ N]\!] = [\![M]\!] \otimes [\![N]\!] \qquad [\![\mathsf{let}\, x \otimes y = M\ \mathsf{in}\ N]\!] = [\![N]\!] \circ ([\![M]\!] \otimes \mathrm{id}_{[\![\Gamma']\!]})$$

$$[\![\mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2)]\!] = ([\![N_1]\!] \otimes [\![N_2]\!])\sigma([\![M]\!] \otimes \mathrm{id}_{[\![\Gamma_1]\!]} \otimes \mathrm{id}_{[\![\Gamma_2]\!]}) \qquad [\![\mathsf{assoc}^{\lhd}_R]\!] = \alpha$$

$$[\![\mathsf{assoc}^{\lhd}_L]\!] = \alpha^{-1} \qquad [\![\mathsf{interch}(M)]\!] = \sigma \circ [\![M]\!] \qquad [\![\mathsf{await}(M)]\!] = [\![M]\!] \qquad [\![\mathsf{expose}(M)]\!] = [\![M]\!]$$

$$[\![|0\rangle]\!] = |0\rangle \qquad [\![U]\!] = U \qquad [\![\mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2]\!] = qif_A \circ ([\![M]\!] \otimes \langle [\![N_1]\!], [\![N_2]\!] \rangle)$$

Figure 6 Degenerate categorical semantics in **Hilb**.

Here, $\Lambda_{X,Y,Z}F\colon X \to (Y \multimap Z)$ is the transpose map of $F\colon X \otimes Y \to Z$, $\mathrm{ev}_{X,Y}\colon (X \multimap Y) \otimes X \to Y$ is the evaluation map, $\sigma\colon X \otimes Y \otimes Z \otimes W \to X \otimes Z \otimes Y \otimes W$ is the isomorphism that swaps the middle two, $\langle f, g\rangle\colon X \to Y \otimes Z$ is the map that is obtained from the universality of the product from $f\colon X \to Y$ and $g\colon X \to Z$.

For the semantics of $\mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2$, we use a linear map $qif_n\colon 2 \otimes (n \oplus_{\mathbf{Hilb}} n) \longrightarrow 2 \otimes n$ for each object n as follows:

$$\begin{array}{rccc} qif_n\colon & 2 \otimes (n \oplus n) & \longrightarrow & 2 \otimes n \\ & |0\rangle \otimes (|\varphi\rangle \oplus |\psi\rangle) & \longmapsto & |0\rangle \otimes |\varphi\rangle \\ & |1\rangle \otimes (|\varphi\rangle \oplus |\psi\rangle) & \longmapsto & |1\rangle \otimes |\psi\rangle\,. \end{array}$$

The following two examples show that the semantics of quantum conditional branching is defined as expected with this map qif_n.

▶ **Example 13.** Let M be $(\mathsf{qif}\ x\ \mathsf{then}\ \mathsf{U}\, y\ \mathsf{else}\ y)$. This term has the type derivation $x, y\colon \mathsf{qbit} \vdash M\colon \mathsf{qbit} \rhd \mathsf{qbit}$, and the semantics are given by the controlled-U gate as $[\![M]\!] = \mathsf{CU}$. ◀

▶ **Example 14.** When $A \in \mathbb{N}^+$, the semantics of the term $x\colon \mathsf{qbit}, f, g\colon A \multimap A \vdash \mathit{SWITCH}\colon \mathsf{qbit} \rhd (A \multimap A)$ we defined in Example 9 coincides with the usual quantum SWITCH [17] described as

$$(\alpha\,|0\rangle + \beta\,|1\rangle) \otimes f \otimes g \longmapsto (\alpha\,|0\rangle \otimes (gf)) + (\beta\,|1\rangle \otimes (fg)).$$ ◀

▶ Remark 15. It is worth noting that a closed monoidal category is a (degenerate) model of IBV, in which $(A \lhd B) \cong (A \otimes B)$. This is in contract to the classical setting: a $*$-autonomous category is not necessarily a model of BV logic. The difference comes from the self-duality of $\lhd$ required only in the classical case. A model of (classical) BV logic satisfying $(A \lhd B) \cong (A \otimes B)$ also satisfies $(A \otimes B) \cong (A \lhd B) \cong \neg((\neg A) \lhd (\neg B)) \cong \neg((\neg A) \otimes (\neg B)) \cong (A \parr B)$, so it is compact closed. We think that this difference in the behavior of $\lhd$ between the intuitionistic and classical settings is interesting, although we cannot discuss it further in this paper. ◀

5 Causal Model of Pure Quantum Computation

Although we have already defined a categorical model of λ^{QIF} in Section 4.4, it does not reflect the causal structure of the language well enough since it is defined in **Hilb** where $\otimes = \lhd = \parr$. This is problematic and does not allow us to analyze our language in detail. For example, it is not easy to prove whether there is no program that increases the total probability of a state as in the paradox we explained in the Introduction.

In this section, we construct a new categorical model of pure quantum computation **CausHilb**, which has enough structure to interpret BV-logic and our language.

5.1 Caus Construction

Kissinger, Uijlen and Simmons [35, 51] have developed a construction of a categorical model of causality, called the *Caus construction*. With the Caus construction, from a compact closed category $\mathcal{C}$ with products and *discarding maps*, called an *additive precausal category*, one can construct a category $\mathbf{Caus}[\mathcal{C}]$ that captures the essence of causal ordering of processes. Here, we briefly review their results.

▶ **Definition 16.** *A compact closed category $\mathcal{C}$ with products is called* additive precausal *if it is equipped with a family of morphisms $\{\dot{\top}_A \in \mathcal{C}(A, I)\}_{A\in\mathcal{C}}$ satisfying the following axioms.*

AP1. $\dot{\top}_{A\otimes B} = \dot{\top}_A \otimes \dot{\top}_B$, $\dot{\top}_I = \mathrm{id}_I$, $\dot{\top}_{A\oplus B} = \dot{\top}_A \oplus \dot{\top}_B$.
AP2. $\dot{\top}_A \circ \dot{\bot}_A$ *is invertible for any* $A \neq 0$, *where* $\dot{\bot}_A := (\dot{\top}_{A^*})^*$.
AP3. For each A, there exists a finite set $\{\rho_i \in \mathcal{C}(I, A)\}_{i=1}^n$ such that $\dot{\top}_A \circ \rho_i = \mathrm{id}_I$ and jointly monic, i.e., $\forall f, g \in \mathcal{C}(A, B).\ (\forall i.\ f \circ \rho_i = g \circ \rho_i) \implies f = g$.
AP4. For the commutative semiring of scalars $R := \mathcal{C}(I, I)$, the addition is cancellative, the canonical preorder induced by addition is total, and $R^\times$ forms a group by multiplication.
AP5. $\forall \pi \in \mathcal{C}(A, I).\ \exists \pi' \in \mathcal{C}(A, I).\ \exists \lambda \in R.\ \pi + \pi' = \lambda \cdot \dot{\top}_A$. ◀

▶ **Definition 17.** *Let $\mathcal{C}$ be an additive precausal category. For a set $c \subseteq \mathcal{C}(I, A)$, we define the set $c^* \subseteq \mathcal{C}(A, I)$ by $\{\pi \in \mathcal{C}(A, I) \mid \forall \rho \in c.\ \pi \circ \rho = \mathrm{id}_I\}$. If $c^{**} = c$ holds, c is called* closed.

A set $c \subseteq \mathcal{C}(I, A)$ is called flat *if there exist invertible scalars μ and θ such that $\mu \cdot \dot{\bot}_A \in c$ and $\theta \cdot \dot{\top}_A \in c^*$.*

The category $\mathbf{Caus}[\mathcal{C}]$ consists of the following: An object is a pair $\mathbf{A} = (A, c_{\mathbf{A}})$ of an object A and a closed and flat set $c_{\mathbf{A}} \subseteq \mathcal{C}(I, A)$. A morphism $f\colon \mathbf{A} \longrightarrow \mathbf{B}$ is a map $f \in \mathcal{C}(A, B)$ such that $\forall \rho \in c_{\mathbf{A}}.\ f \circ \rho \in c_B$. ◀

The category $\mathbf{Caus}[\mathcal{C}]$ is a full subcategory of $\mathbf{T}(\mathcal{C})$ where the objects are pairs $(A, c_{\mathbf{A}})$ of an object A and a closed set $c_{\mathbf{A}}$. In [32], this category $\mathbf{T}(\mathcal{C})$ is called the tight category induced by focussed orthogonality for the singleton set $\{\mathrm{id}_I\} \subseteq R$, so it is $*$-autonomous. By proving flatness is closed under all $*$-autonomous structures, one can prove the following.

▶ **Proposition 18** ([35])**.** $\mathbf{Caus}[\mathcal{C}]$ *is a $*$-autonomous category with products, where the monoidal structure and dual are defined as follows. The obvious forgetful functor $\mathbf{Caus}[\mathcal{C}] \longrightarrow \mathcal{C}$ preserves all structures on the nose.*

$$\mathbf{I} := (I, \{\mathrm{id}_I\}) \qquad \mathbf{A}^* := (A, c_{\mathbf{A}}^*)$$
$$\mathbf{A} \otimes \mathbf{B} := (A \otimes B, c_{A\otimes B} := \{\rho_A \otimes \rho_B \mid \rho_A \in c_{\mathbf{A}}, \rho_B \in c_{\mathbf{B}}\}^{**})$$ ◀

▶ **Example 19.** **CPM** is additive precausal, thus it admits the **Caus** construction. ◀

5.2 Our Categorical Model: CausHilb

The Caus construction is indeed a remarkable construction. It is not only $*$-autonomous, but can also interpret derivations in BV-logic. Moreover, $\mathbf{Caus}[\mathbf{CPM}]$ does provide a categorical model that captures causal relationships in quantum computing. However, it does not provide a model for our language. In particular, the semantics of quantum conditional qif in our language is defined using the additive enrichment in **Hilb** corresponding to

quantum superposition, which is not in **CPM** or **Caus**[**CPM**], where the additive enrichment corresponds to probabilistic mixing.

One might then consider applying the Caus construction to **Hilb**, but this does not work since **Hilb** is not additive precausal. In fact, the requirement for a category to be additive precausal is quite strong. For example, AP4. asks the scalars to be totally ordered like $\mathbb{R}^+$, which is not the case for $\mathbf{Hilb}(1,1) = \mathbb{C}$.

Instead of directly applying the Caus construction to **Hilb** or **CPM**, we define a category by gluing together the structure in **Hilb** and **Caus**[**CPM**] as follows.

▶ **Definition 20.** *The category* **CausHilb** *consists of the following: Objects of* **CausHilb** *are given by pairs* (n, c) *of a natural number* n *and a closed and flat set* $c \subseteq \mathbf{CPM}(1,(n))$*. Morphisms from* (n, c) *to* (m, c') *are linear maps* $f \in \mathbf{Hilb}(n, m)$ *such that* $\forall \rho \in c, (\iota f)\circ\rho \in c'$*.*

There is a canonical forgetful functor $\mathbf{CausHilb} \longrightarrow \mathbf{Hilb}$*, and a functor* $\mathbf{CausHilb} \longrightarrow \mathbf{Caus}[\mathbf{CPM}]$ *which maps* (n, c) *to* $((n), c)$ *and* f *to* $\iota(f)$*.*[3] ◀

In fact, there is an alternative definition for **CausHilb**.

▶ **Lemma 21.** *The following diagram is a pullback square in* **Cat**.

$$\begin{array}{ccc} \mathbf{CausHilb} & \longrightarrow & \mathbf{Caus}[\mathbf{CPM}] \\ \big\downarrow \lrcorner & & \big\downarrow U \\ \mathbf{Hilb} & \xrightarrow{\quad\iota\quad} & \mathbf{CPM} \end{array} \qquad ◀ \qquad (2)$$

▶ **Proposition 22.** **CausHilb** *is* $*$*-autonomous and the functors* $\mathbf{CausHilb} \longrightarrow \mathbf{Hilb}$ *and* $\mathbf{CausHilb} \longrightarrow \mathbf{Caus}[\mathbf{CPM}]$ *preserves all* $*$*-autonomous structures on the nose.*

Proof. The category **AutCat** of $*$-autonomous categories and functors that preserve every structure on the nose is monadic over **Cat** [11]. So the forgetful functor $\mathbf{AutCat} \longrightarrow \mathbf{Cat}$ creates limits. Since the cospan in the diagram (2) is a cospan in **AutCat**, **CausHilb** is also $*$-autonomous. ◀

From the previous proposition, we can easily compute the $*$-autonomous structure in **CausHilb**. For example, the monoidal unit is $\mathbf{I} := (1, \{\mathrm{id}_I\})$, and the monoidal product of objects can be computed in **Caus**[**CPM**], *i.e.*, an object $(n, c_\mathbf{A}) \in \mathbf{CausHilb}$ can be identified with an object of $((n), c_\mathbf{A}) \in \mathbf{Caus}[\mathbf{CPM}]$, and the monoidal product can be computed via this identification. Note that, even though we can compute the structure in such a way, $\mathbf{CausHilb} \longrightarrow \mathbf{Caus}[\mathbf{CPM}]$ is not full or faithful and the categories themselves are very different.

▶ **Proposition 23.** **CausHilb** *has all finite products and coproducts, preserved by the forgetful functor* $\mathbf{CausHilb} \longrightarrow \mathbf{Hilb}$.

Proof. Let $\mathbf{A} = (A, c_\mathbf{A})$ and $\mathbf{B} = (B, c_\mathbf{B})$ be objects of **CausHilb**. Without loss of generality, we assume $\text{⏚}_A \in c_\mathbf{A}$ and $\text{⏚}_B \in c_\mathbf{B}$. The product of the objects is given by

$$\mathbf{A} \times \mathbf{B} := (A \oplus_{\mathbf{Hilb}} B, \{\rho \mid (\iota p_A)\rho \in c_\mathbf{A}, (\iota p_B)\rho \in c_\mathbf{B}\}),$$

where $p_\mathbf{A}$ is the projection $A \oplus_{\mathbf{Hilb}} B \longrightarrow A$ in **Hilb**. Since **CausHilb** is self-dual, it also admits coproducts $\mathbf{A} \sqcup \mathbf{B} := (\mathbf{A}^* \times \mathbf{B}^*)^*$. ◀

[3] Abusing the notation, we also write the functor $\mathbf{CausHilb} \longrightarrow \mathbf{Caus}[\mathbf{CPM}]$ as ι.

5.3 BV-category

We now show that our model forms a BV-category, in which BV-logic has interpretation. BV-category was first defined in [12], and later studied in [28] in detail. We first sketch the definition of BV-categories, asking the reader to consult the other papers for details.

▶ **Definition 24** ([12, 28]). *A BV-category is a category with three monoidal structures $\otimes$, $\parr$, and $\triangleleft$ with isomorphic units $I \cong I^* \cong J$, with the following structures:*

- *a $*$-autonomous isoMIX structure with respect to $\otimes$ and $\parr$,*
- *a self-dual structure $(A \triangleleft B)^* \cong A^* \triangleleft B^*$,*
- *a duoidal category structure with respect to $\otimes$ and $\triangleleft$,* i.e., *interchange natural transformation $\zeta\colon (A \triangleleft C) \otimes (B \triangleleft D) \longrightarrow (A \otimes B) \triangleleft (C \otimes D)$ that is compatible with the monoidal structures,*
- *and another duoidal category structure with respect to $\triangleleft$ and $\parr$, which derives as the dual of that for $\otimes$ and $\triangleleft$.* ◀

To distinguish three associators and left/right unitors for BV-categories, we call them $\alpha^\bullet$, $\ell^\bullet$ and $r^\bullet$ for each $\bullet \in \{\otimes, \parr, \triangleleft\}$.

▶ Remark 25. Although the BV-categories are expected to serve as a sound model of BV-logic in a certain appropriate sense, this remains a partially open problem. Recently, Acclavio et al. [4] have proven that a *strong BV-category*, which is a special BV-category with a faithful embedding into a compact closed category that satisfies some coherence axioms, does model BV-logic in an appropriate sense. Our model and the forgetful functor **CausHilb** $\longrightarrow$ **Hilb** defines a strong BV-category, so it is a model of BV-category. ◀

Rather than directly proving that **CausHilb** is a BV-category by spelling out all coherence diagrams and risking overlooking something, we shall prove this by showing that pullbacks in **Cat** preserve the BV-category structure. For this purpose, we first state that the other three categories **Hilb**, **CPM**, and **Caus**[**CPM**] are BV-categories.

▶ **Proposition 26.** *A compact closed category is a BV-category where $\otimes = \triangleleft = \parr$.* ◀

▶ **Theorem 27** ([51]). **Caus**$[\mathcal{C}]$ *is a BV-category and all structures are preserved on the nose by the forgetful functor* **Caus**$[\mathcal{C}] \longrightarrow \mathcal{C}$. *The object $\mathbf{A} \triangleleft \mathbf{B}$ is defined by $(A \otimes B, c_{\mathbf{A} \triangleleft \mathbf{B}})$ where*

$$c_{\mathbf{A} \triangleleft \mathbf{B}} := \left\{ h \in \mathcal{C}(I, A \otimes B) \;\middle|\; \forall \pi, \pi' \in c_{\mathbf{B}}^*, (\mathrm{id}_A \otimes \pi) \circ h = (\mathrm{id}_A \otimes \pi') \circ h \in c_{\mathbf{A}} \right\}$$ ◀

To prove that pullbacks preserve BV-category structures, we use a consequence of categorical universal algebra.

▶ **Lemma 28.** *The category of BV-categories and functors that strictly preserve every structure on the nose is finitary monadic over* **Cat**.

Sketch of proof. The theory of categories $\mathbb{T}_{\mathrm{Cat}}$ can be given as a partial Horn theory [43] of two sorts C_0 (objects) and C_1 (morphisms) with function symbols such as $\mathrm{dom}\colon C_1 \to C_0$. The theory of BV-categories can be given by adding more function symbols such as $\triangleleft_0\colon C_0, C_0 \to C_0$ (the action of the functor $\triangleleft$ to objects) or $\zeta\colon C_0, C_0, C_0, C_0 \to C_1$ (the natural transformation ζ) and many equations corresponding to the coherence diagrams. Therefore, the theory of BV-categories can be written as a $\mathbb{T}_{\mathrm{Cat}}$-relative algebraic theory in [34], so the category of BV-categories is monadic over **Cat**. ◀

▶ **Corollary 29.** **CausHilb** *is a BV-category, and all structures are preserved on the nose by the functors* **CausHilb** $\longrightarrow$ **Hilb** *and* **CausHilb** $\longrightarrow$ **Caus**[**CPM**].

Proof. From Lemma 28, the forgetful functor creates the pullback (2) in **Cat**. ◀

The BV-category structure of **CausHilb** on objects can also be computed via the identification through ι. That is, since ι is injective on objects, $\mathbf{A} \lhd \mathbf{B}$ in **CausHilb** can be computed by computing $(\iota\mathbf{A}) \lhd (\iota\mathbf{B})$ in **Caus**[**CPM**]. On the other hand, for any morphisms $f \in \mathbf{CausHilb}(\mathbf{A}, \mathbf{B})$ and $g \in \mathbf{CausHilb}(\mathbf{A}', \mathbf{B}')$, since the forgetful functor to **Hilb** preserves the structures, the underlying morphisms of $f \otimes g$, $f \parr g$, and $f \lhd g$ in **CausHilb** are all $f \otimes g \in \mathbf{Hilb}(A \otimes A', B \otimes B')$.

5.4 First-order Objects in CausHilb

In a category constructed by the **Caus** construction [35, 51, 52], it has been shown that the objects called *first-order* are particularly crucial. These are the objects that can be written as $(A, \{\bar{+}_A\}^*) \in \mathbf{Caus}[\mathcal{C}]$. Among all objects $(A, c_{\mathbf{A}})$ whose underlying object is $A \in \mathcal{C}$, such first-order objects are the ones that have the maximal set $c_{\mathbf{A}} \subseteq \mathcal{C}(I, A)$. Since an intuitive meaning of $c_{\mathbf{A}}$ is the collection of causally coherent states, first-order objects are the most unrestricted, simplest objects. For example, in the category **Caus**[**CPM**], the full subcategory of first-order objects defines the category **CPTP**, the category of completely positive trace preserving maps, which is a natural model of first-order quantum computation.

The most important aspect of first-order objects is that, it models the first-order propositions in the causal logic. Indeed, in [52], Simmons and Kissinger have shown that **Caus**[$\mathcal{C}$] defines a complete model of causal logic for any non-trivial $\mathcal{C}$.

In our model **CausHilb**, in addition to the BV-category structure, we can also inherit first-order objects from **Caus**[**CPM**]. We also see that these objects indeed capture first-order pure quantum computation.

▶ **Definition 30.** *An object* $(n, c_{\mathbf{A}})$ *in* **CausHilb** *is called* first-order *if* $c_{\mathbf{A}} = \{\bar{+}_{(n)}\}^*$.[4] ◀

▶ **Theorem 31.** *The full subcategory of first-order objects in* **CausHilb** *is isomorphic to the category* **Isom** *of isometries,* i.e.*, the category whose objects are natural numbers and whose morphisms* $n \longrightarrow m$ *are isometries* $\mathbb{C}^n \longrightarrow \mathbb{C}^m$.

Proof. Let us denote the first-order object $(n, \{\bar{+}_{(n)}\}^*)$ as $\mathbf{n}$. A morphism $\mathbf{n} \longrightarrow \mathbf{m}$ in **CausHilb** is a linear map $|\psi\rangle \longmapsto V|\psi\rangle$ such that $\rho \longmapsto V\rho V^\dagger$ defines a trace-preserving map. Such V is an isometry. ◀

▶ Remark 32. As in **Caus**[$\mathcal{C}$], first-order objects are closed under $\otimes$, and there, all three monoidal products coincide, *i.e.*, $\mathbf{n} \otimes \mathbf{m} = \mathbf{n} \lhd \mathbf{m} = \mathbf{n} \parr \mathbf{m}$. Also, the dual of a first-order object is first-order if and only if it is $\mathbf{I}$.

On the other hand, in **Caus**[$\mathcal{C}$], it is shown that first-order objects are closed under coproducts, but this is not true in **CausHilb**. In our model **CausHilb**, first-order objects are closed under neither products nor coproducts. Thus, for example, the semantics of the qbit type, which will be defined to be the first-order object $\mathbf{2}$ in the next section, is not a product or coproduct of the units $\mathbf{I}$. ◀

We show that **CausHilb** can also model first-order propositions by these first-order objects. The key observation here is that the embedding functor **CausHilb** $\longrightarrow$ **Caus**[**CPM**] is

[4] Instead of this definition, we can also define first-order objects as objects whose $c^*_{\mathbf{A}}$ is a singleton. Since such objects are isomorphic to one of $(n, \{\bar{+}\}^*)$, these two definitions define an equivalent full subcategory of **CausHilb**.

conservative, *i.e.*, it reflects isomorphisms. From this fact, we can state the following: for any proof $A \vdash B$ in BV-logic, we can give a semantics of the proof in **CausHilb** and **Caus**[**CPM**] by specifying an object in **CausHilb** for each atomic proposition that occurs in A or B. These semantics are related by the embedding functor. Then, if the semantics in **Caus**[**CPM**] happened to be an isomorphism, the semantics of the proof in **CausHilb** must also be an isomorphism. This idea can be used in the next proposition.

▶ **Proposition 33.** *In the category* **CausHilb***, we have the following natural transformation as part of the duoidal structure* $(\otimes, \lhd)$*.*

$$\zeta_{A,B,C,D} \colon (A \lhd C) \otimes (B \lhd D) \longrightarrow (A \otimes B) \lhd (C \otimes D).$$

From this ζ*, by assuming* $B = C = \mathbf{I}$*, we obtain the following* $\xi_{A,B}$*. Also, by considering* ζ^*_{B,C^*,I,A^*}*, we obtain the following* $\gamma_{B,C,A}$*.*

$$\xi_{A,B} \colon A \otimes B \longrightarrow A \lhd B \qquad \gamma_{B,C,A} \colon (B \multimap C) \lhd A \longrightarrow B \multimap (C \lhd A)$$

When A *is first-order, these morphisms are isomorphisms.*

Proof. The converse $A \lhd B \vdash A \otimes B$ is derivable in the causal logic when A is a first-order proposition, and the semantics of the proof gives the inverse of $\iota(\xi_{A,B})$. Since $\iota(\xi_{A,B})$ is an isomorphism in **Caus**[**CPM**], $\xi_{A,B}$ is also an isomorphism. The same proof applies to γ. ◀

▶ Remark 34. We can also prove this proposition easily by directly checking the existence of the inverse, but our approach is much easier to be generalized to any other such morphisms. ◀

6 Categorical Semantics with Causal Structure

We now give the semantics of λ^{QIF} in **CausHilb** and study its properties. We prove the full abstraction theorem in Section 6.2, and we study the definability in Section 6.3.

6.1 Definition of Categorical Semantics

In this section, we define a refined categorical semantics $[\![-]\!]_{\mathbf{CausHilb}}$ of λ^{QIF} in **CausHilb**. This categorical semantics has additional information about causality, and by forgetting it, we can recover the original degenerate semantics which we defined in Section 4.4. More concretely, through the forgetful functor $U : \mathbf{CausHilb} \longrightarrow \mathbf{Hilb}$, we can write the semantics in **Hilb** as $U[\![-]\!]_{\mathbf{CausHilb}} = [\![-]\!]_{\mathbf{Hilb}}$.

The semantics of types are given by the following, relating each syntactic connective with its corresponding categorical structure.

$$[\![n]\!] = (n, \{\dot{\top}_{(n)}\}^*) \qquad [\![\bot]\!] = \mathbf{I} = (1, \{\mathrm{id}_{(1)}\}) \qquad [\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$$
$$[\![A \lhd B]\!] = [\![A]\!] \lhd [\![B]\!] \qquad [\![A \multimap B]\!] = [\![A]\!] \multimap [\![B]\!]$$

For every type, the underlying object in **Hilb**, *i.e.*, the natural number representing the dimension, coincides with the semantics in **Hilb**. It is notable that, the semantics of first-order types are given by first-order objects.

For each term $\Gamma \vdash M \colon A$, its categorical semantics is defined in Figure 7 as a map $[\![M]\!]_{\mathbf{CausHilb}} \colon [\![\Gamma]\!] \longrightarrow [\![A]\!]$ in **CausHilb**.

Similarly to the semantics in Section 4.4, the semantics of the terms in linear lambda calculus are defined in a standard way in the monoidal closed category **CausHilb**. We explain the semantics for the following non-trivial ones.

$$[\![x\colon A]\!] = \mathrm{id}_{[\![A]\!]} \qquad [\![()]\!] = \mathrm{id}_{\mathbf{I}} \qquad [\![\lambda x.\, M]\!] = \Lambda_{[\![\Gamma]\!],[\![A]\!],[\![B]\!]}[\![M]\!]$$

$$[\![M\ N]\!] = \mathrm{ev}_{[\![A]\!],[\![B]\!]}([\![M]\!] \otimes [\![N]\!]) \qquad [\![M \otimes N]\!] = [\![M]\!] \otimes [\![N]\!]$$

$$[\![\mathsf{let}\ \_ = M\ \mathsf{in}\ N]\!] = \ell^{\otimes}([\![M]\!] \otimes [\![N]\!]) \qquad [\![\mathsf{let}\, x \otimes y = M\ \mathsf{in}\ N]\!] = [\![N]\!] \circ ([\![M]\!] \otimes \mathrm{id}_{[\![\Gamma']\!]})$$

$$[\![M \lhd N]\!] = \xi_{[\![A]\!],[\![B]\!]} \circ ([\![M]\!] \otimes [\![N]\!])$$

$$[\![\mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2)]\!] = ([\![N_1]\!] \lhd [\![N_2]\!]) \circ \zeta_{[\![A]\!],[\![B]\!],[\![\Gamma_1]\!],[\![\Gamma_2]\!]} \circ ([\![M]\!] \otimes \xi_{[\![\Gamma_1]\!],[\![\Gamma_2]\!]})$$

$$[\![\mathsf{assoc}^{\lhd}_{R}]\!] = \alpha^{\lhd} \qquad [\![\mathsf{assoc}^{\lhd}_{L}]\!] = (\alpha^{\lhd})^{-1} \qquad [\![\mathsf{interch}(M)]\!] = \zeta_{[\![A]\!],[\![B]\!],[\![C]\!],[\![D]\!]} \circ [\![M]\!]$$

$$[\![\mathsf{await}(M)]\!] = \xi^{-1}_{[\![A]\!],[\![B]\!]} \circ [\![M]\!] \qquad [\![\mathsf{expose}(M)]\!] = \gamma^{-1}_{[\![B]\!],[\![C]\!],[\![A]\!]} \circ [\![M]\!] \qquad [\![|0\rangle]\!] = |0\rangle$$

$$[\![U]\!] = U \qquad [\![\mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2]\!] = qif_A \circ ([\![M]\!] \otimes \langle [\![N_1]\!], [\![N_2]\!] \rangle)$$

Figure 7 Categorical semantics of terms in **CausHilb**

For the term $\Gamma, \Gamma' \vdash M \lhd N\colon A \lhd B$, the semantics is given by a morphism $[\![\Gamma]\!] \otimes [\![\Gamma']\!] \longrightarrow [\![A]\!] \lhd [\![B]\!]$. To obtain this morphism from $[\![M]\!]$ and $[\![N]\!]$, we can first take the $\otimes$-monoidal product of them to obtain a map $[\![\Gamma]\!] \otimes [\![\Gamma']\!] \longrightarrow [\![A]\!] \otimes [\![B]\!]$, and then post-compose the map $\xi_{[\![A]\!],[\![B]\!]}\colon [\![A]\!] \otimes [\![B]\!] \longrightarrow [\![A]\!] \lhd [\![B]\!]$.

For the term $\Gamma_0, \Gamma_1, \Gamma_2 \vdash \mathsf{let}\, x \lhd y = M\ \mathsf{in}\ (N_1 \lhd N_2)$, the semantics is given by a morphism of type $[\![\Gamma_0]\!] \otimes [\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!] \longrightarrow [\![C]\!] \lhd [\![D]\!]$. This can be obtained as follows:

$$\begin{aligned}[\![\Gamma_0]\!] \otimes ([\![\Gamma_1]\!] \otimes [\![\Gamma_2]\!]) &\xrightarrow{[\![M]\!] \otimes \xi_{[\![\Gamma_1]\!],[\![\Gamma_2]\!]}} ([\![A]\!] \lhd [\![B]\!]) \otimes ([\![\Gamma_1]\!] \lhd [\![\Gamma_2]\!]) \\ &\xrightarrow{\zeta_{[\![A]\!],[\![\Gamma_1]\!],[\![B]\!],[\![\Gamma_2]\!]}} ([\![A]\!] \otimes [\![\Gamma_1]\!]) \lhd ([\![B]\!] \otimes [\![\Gamma_2]\!]) \\ &\xrightarrow{[\![N_1]\!] \lhd [\![N_2]\!]} [\![C]\!] \lhd [\![D]\!].\end{aligned}$$

For the term $\mathsf{qif}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_2$, to define the semantics, it suffices to show that the morphism qif_A lifts to **CausHilb**.

▶ **Proposition 35.** *For each object* $\mathbf{A} \in$ **CausHilb**, *the map* qif_A *in* **Hilb** *defines a map* $[\![\mathsf{qbit}]\!] \otimes (\mathbf{A} \times \mathbf{A}) \longrightarrow [\![\mathsf{qbit}]\!] \rhd \mathbf{A}$ *in* **CausHilb**. ◀

6.2 Full Abstraction

Though we do not provide an operational semantics for λ^{QIF}, we can still discuss full abstraction for our language. The following term H_d is critical in the proof.

▶ **Lemma 36.** *For each* $d \in \mathbb{N}^+$, *there is a program* $\vdash H_d\colon d \otimes d$ *whose semantics is* $\frac{1}{\sqrt{d}}\eta_d$ *where* η_d *is the unit* $\eta_d\colon I \longrightarrow d \otimes d$ *of the compact closed structure in* **Hilb**. ◀

▶ **Lemma 37.** *For any type* A, *there are some natural numbers* n_A, m_A, ℓ_A *and two terms* $x_A\colon A \vdash M_A\colon n_A$ *and* $y_A\colon m_A \vdash N_A\colon A \lhd \ell_A$ *such that* $[\![M_A]\!]$ *is injective in* **Hilb** *and* $(\mathrm{id}_A \otimes \varepsilon_\ell)([\![N]\!] \otimes \mathrm{id}_\ell)\colon m \otimes \ell \longrightarrow A$ *is surjective in* **Hilb**.

Sketch of Proof. Ignoring some variable renamings and type equivalences, the terms can be recursively defined as follows:

- $M_d = M_\perp = x$, $N_d = N_\perp = y$,

- $M_{A\otimes B} = M_A \otimes M_B$, $N_{A\otimes B} = \mathsf{interch}(N_A \otimes N_B)$,
- $M_{A\lhd B} = \mathsf{let}\, x_A \lhd x_B = x \,\mathsf{in}\, (M_A \lhd M_B)$,
  $N_{A\lhd B} = \mathsf{let}\, (a \otimes b) \lhd z = \mathsf{interch}(N_A \otimes N_B) \,\mathsf{in}\, (a \lhd b) \lhd z$,
- For the function type, we can define the terms as following. Note that there is a equivalence of types $A \multimap ((B \lhd \ell_B) \otimes n_A) \equiv (A \multimap B) \lhd (\ell_B \otimes n_A)$ via the term expose.

$$\begin{aligned} M_{A\multimap B} = {} & \mathsf{let}\, y_A \otimes z^{m_A} = H_{m_A} \,\mathsf{in}\, \mathsf{let}\, z^A \lhd z^{\ell_A} = N_A \,\mathsf{in} \\ & ((\mathsf{let}\, x_B = x\, z^A \,\mathsf{in}\, M_B) \lhd z^{\ell_B}) \otimes z^{m_A}, \\ N_{A\multimap B} = {} & \lambda x_A.\, N_B \otimes M_A. \end{aligned}$$

◀

▶ **Theorem 38** (Full abstraction). *Let $\Gamma \vdash M : A$ and $\Gamma \vdash N : A$. Then, $[\![C[M]]\!] = [\![C[N]]\!]$ for any context $C[-]$ of type $d \in \mathbb{N}^+$, if and only if $[\![M]\!] = [\![N]\!]$.*

Proof. From the compositionality of the semantics, the if part is trivial. We prove the converse. Let $\Gamma = \vec{x_i} : \vec{B_i}$. Then, using N_{B_i} and $\mathsf{interch}$, we obtain a term L of type $\vec{y_i} : \vec{m_{B_i}} \vdash L : (\bigotimes B_i) \lhd (\prod \ell_{B_i})$. Now, we define a context $D[-]$ as follows:

$$\vec{y_i} : \vec{m_{B_i}} \vdash \mathsf{let}\, \vec{x_i} \lhd z = L \,\mathsf{in}\, (\mathsf{let}\, x_A = [-] \,\mathsf{in}\, M_A) \lhd z : n_A \lhd \prod \ell_{B_i}.$$

From Lemma 37, $[\![D[-]]\!]$ defines an injective map $\mathbf{Hilb}([\![\Gamma]\!], [\![A]\!]) \longrightarrow \mathbf{Hilb}(\prod_i m_{B_i}, n_A \times \prod_i \ell_{B_i})$. For any map $f, g \in \mathbf{Hilb}(n, m)$, if $f \neq g$, there exists a vector $|\psi\rangle$ of length 1 such that $f|\psi\rangle \neq g|\psi\rangle$. Therefore, if $[\![M]\!] \neq [\![N]\!]$, there exists some unitary map $\mathsf{U} : \prod m_{B_i} \longrightarrow \prod m_{B_i}$ that makes the context $C[-] := \mathsf{let}\, \vec{y_i} = \mathsf{U}\,|0\rangle \,\mathsf{in}\, D[-]$ satisfy $[\![C[M]]\!] \neq [\![C[N]]\!]$. ◀

6.3 Causality and Definability

In this section, we study the class of maps and supermaps that can be described in λ^{QIF}. In particular, we prove that every term is causally consistent in the sense that no paradox described in the Introduction occurs in λ^{QIF}. We do this by studying the definability of the semantics taken in **CausHilb**.

First, for first-order maps, we can prove the following:

▶ **Proposition 39.** *The semantics of every term $x : n \vdash M : m$ is an isometry. Conversely, every map in $\mathbf{Isom}(n, m)$ is definable by a term $x : n \vdash M : m$.*

Proof. It follows from $\mathbf{CausHilb}(n, m) = \mathbf{Isom}(n, m)$ proved in Theorem 31. For the definability, we can define each isometry by a term $\mathsf{U}(x \otimes |0\rangle)$ for some unitary $\mathsf{U} : m \multimap m$. ◀

In particular, this fact shows that our language can avoid the paradox regarding the quantum conditionals we presented in the Introduction.

We can also characterize all pure supermaps:

▶ **Theorem 40.** *A supermap is pure if and only if it can be represented as ιf for some $f : \bigotimes_i (\mathbf{n_i} \multimap \mathbf{m_i}) \multimap \mathbf{n} \multimap \mathbf{m}$ in **CausHilb**. In particular, all definable supermaps $\bigotimes_i (n_i \multimap m_i) \multimap n \multimap m$ in our language are pure, and the OCB process [41] is not definable in λ^{QIF} since it is not pure.*

Proof. Sufficiency follows from Theorem 31 and a similar observation as in Example 11. We prove the necessity. A similar proof can be found in [56]. For each pure supermap $g : \bigotimes_i (\mathcal{L}(\mathbb{C}^{2^{n_i}}) \multimap \mathcal{L}(\mathbb{C}^{2^{m_i}})) \longrightarrow \mathcal{L}(\mathbb{C}^{2^n}) \multimap \mathcal{L}(\mathbb{C}^{2^m})$, by partially applying $\mathsf{swap}_i : n_i \otimes m_i \longrightarrow m_i \otimes n_i$, we obtain an isometry of type $n \otimes_i m_i \longrightarrow m \otimes_i n_i$. Since the original g can be recovered by composing unit and counit of the compact closed structure, and all these maps are in the image of the functor $\iota : \mathbf{Hilb} \longrightarrow \mathbf{CPM}$, the map g itself is in the image of ι. Let $\iota f = g$, then since $f \in \mathbf{Hilb}$ maps isometries to isometry, it is a map in **CausHilb**. ◀

Furthermore, using some facts that have been proven in the quantum community, we can extend the result to the supermaps that take at most two inputs:

▶ **Proposition 41.** *Every map of type* $(\mathbf{n} \multimap \mathbf{m}) \multimap (\mathbf{n}' \multimap \mathbf{m}')$ *in* **CausHilb** *is definable.*

Proof. For single input supermaps (deterministic supermaps), in [15, Theorem 1], it is shown that all such maps can be written as $\lambda f.\mathsf{V} \circ (f \otimes \mathrm{id}) \circ \mathsf{U} \circ (\mathrm{id} \otimes |0\rangle)$. ◀

To discuss supermaps with two inputs, we extend our language by extending the syntax of qif to qif $M \geq m$ then $x \to N_1$ else $y \to N_0$. This represents a coherent branching where, it branches to N_0 if the control qubit M is $|0\rangle, \ldots, |m-1\rangle$, and it branches to N_1 if it is in between $|m\rangle, \ldots |n+m-1\rangle$. The typing rule is now modified as follows.

$$\frac{\Gamma \vdash M : n+m \qquad \Gamma', x_0 : m \vdash N_0 : m \rhd A \qquad \Gamma', x_1 : n \vdash N_1 : n \rhd A}{\Gamma, \Gamma' \vdash \mathsf{qif}\ M \geq m\ \mathsf{then}\ x_1 \to N_1\ \mathsf{else}\ x_1 \to N_0 : (n+m) \rhd A}$$

The original qif M then N_1 else N_2 can be recovered by qif $M \geq 1$ then $x \to (x; N_1)$ else $y \to (y; N_0)$ when $m = n = 1$. The semantics is defined in the Appendix.

▶ **Proposition 42.** *In the language extended by generalizing* qif *that takes a qudit as its condition, every map* $(\mathbf{n} \multimap \mathbf{m}) \multimap (\mathbf{n} \multimap \mathbf{m}) \multimap (\mathbf{n}' \multimap \mathbf{m}')$ *in* **CausHilb** *is definable.*

Proof. For supermaps with two inputs (bipartite supermaps), in [56, Theorem 5], it was shown that such maps can be written by a direct sum of two causally ordered processes. That is, there is a decomposition of the supermap f

$$f(g,h) = \mathsf{V} \circ (f_1(g,h) \oplus f_2(g,h)) \circ \mathsf{U}(\mathrm{id} \otimes |0\rangle)$$

for some unitaries U, V and quantum combs $f_i : (n \multimap m) \multimap (n \multimap m) \multimap n_i \multimap m_i$ which can be realized by a composition of isometries. Since this $\oplus$ can be represented with a generalized qif, this is definable. ◀

▶ Remark 43. However, we conjecture that for supermaps that take three or more inputs, there are some maps which are not definable in our language. This is because it is known that there are some tripartite supermaps that break causal inequality [6], which do not seem to be describable only with qif. ◀

7 Conclusion

We have developed a higher-order quantum programming language equipped with a type system based on causal logic, together with its categorical semantics. In the presence of quantum conditional branching controlling functions, a naïve linear type discipline cannot avoid some physically-unrealizable programs. In this paper, we have shown that tools from pomset/BV/causal logics [47, 25, 52], namely the seq-connective $\lhd$ and the concept of first-order types, resolve this issue. While it has been known that semantic models of quantum computation can exhibit structures of causality-aware logics, our results strengthen this connection by showing that some ideas from pomset/BV/causal logics are in fact indispensable in higher-order quantum computation.

References

1. Alastair A. Abbott, Christina Giarmatzi, Fabio Costa, and Cyril Branciard. Multipartite causal correlations: Polytopes and inequalities. *Phys. Rev. A*, 94:032131, Sep 2016. URL: https://link.aps.org/doi/10.1103/PhysRevA.94.032131, doi:10.1103/PhysRevA.94.032131.
2. Matteo Acclavio and Giulia Manara. Proofs as execution trees for the π-calculus, 2025. URL: https://arxiv.org/abs/2411.08847, arXiv:2411.08847.
3. Matteo Acclavio and Lutz Straßburger. Intuitionistic BV. In Gian Luca Pozzato and Tarmo Uustalu, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 34th International Conference, TABLEAUX 2025, Reykjavik, Iceland, September 27-29, 2025, Proceedings*, volume 15980 of *Lecture Notes in Computer Science*, pages 414–432. Springer, 2025. doi:10.1007/978-3-032-06085-3_22.
4. Matteo Acclavio, Lutz Straßburger, and Vladimir Zamdzhiev. Proof identity and categorical models of BV. In Frank Pfenning, editor, *11th International Conference on Formal Structures for Computation and Deduction (FSCD 2026)*, volume 378 of *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. To appear. URL: https://arxiv.org/abs/2604.25501, arXiv:2604.25501.
5. Mateus Araújo, Fabio Costa, and Časlav Brukner. Computational advantage from quantum-controlled ordering of gates. *Physical Review Letters*, 113(25):250402, dec 2014. doi:10.1103/physrevlett.113.250402.
6. Mateus Araújo, Adrien Feix, Miguel Navascués, and Caslav Brukner. A purification postulate for quantum mechanics with indefinite causal order. *Quantum*, 1:10, 2017. URL: https://doi.org/10.22331/q-2017-04-26-10, doi:10.22331/Q-2017-04-26-10.
7. Mateus Araújo, Cyril Branciard, Fabio Costa, Adrien Feix, Christina Giarmatzi, and Časlav Brukner. Witnessing causal nonseparability. *New Journal of Physics*, 17(10):102001, oct 2015. doi:10.1088/1367-2630/17/10/102001.
8. Michael Barr. *-autonomous categories and linear logic. *Math. Struct. Comput. Sci.*, 1(2):159–178, 1991. doi:10.1017/S0960129500001274.
9. Ämin Baumeler, Adrien Feix, and Stefan Wolf. Maximal incompatibility of locally classical behavior and global causal order in multiparty scenarios. *Phys. Rev. A*, 90:042106, Oct 2014. URL: https://link.aps.org/doi/10.1103/PhysRevA.90.042106, doi:10.1103/PhysRevA.90.042106.
10. Ämin Baumeler and Stefan Wolf. The space of logically consistent classical processes without causal order. *New Journal of Physics*, 18(1):013036, jan 2016. doi:10.1088/1367-2630/18/1/013036.
11. R. Blackwell, G.M. Kelly, and A.J. Power. Two-dimensional monad theory. *Journal of Pure and Applied Algebra*, 59(1):1–41, 1989. URL: https://www.sciencedirect.com/science/article/pii/0022404989901606, doi:10.1016/0022-4049(89)90160-6.
12. Richard Blute, Prakash Panangaden, and Sergey Slavnov. Deep inference and probabilistic coherence spaces. *Applied Categorical Structures*, 20(3):209–228, 2012. doi:10.1007/s10485-010-9241-0.
13. Cyril Branciard, Mateus Araújo, Adrien Feix, Fabio Costa, and Časlav Brukner. The simplest causal inequalities and their violation. *New Journal of Physics*, 18(1):013008, dec 2015. doi:10.1088/1367-2630/18/1/013008.
14. Lutz Straß Burger and Alessio Guglielmi. A system of interaction and structure iv: The exponentials and decomposition. *ACM Trans. Comput. Logic*, 12(4), July 2011. doi:10.1145/1970398.1970399.
15. G. Chiribella, G. M. D'Ariano, and P. Perinotti. Transforming quantum operations: Quantum supermaps. *Europhysics Letters*, 83(3):30004, jul 2008. doi:10.1209/0295-5075/83/30004.
16. Giulio Chiribella, Manik Banik, Some Sankar Bhattacharya, Tamal Guha, Mir Alimuddin, Arup Roy, Sutapa Saha, Sristy Agrawal, and Guruprasad Kar. Indefinite causal order enables perfect quantum communication with zero capacity channels. *New Journal of Physics*, 23(3):033039, March 2021. doi:10.1088/1367-2630/abe7a0.

17 Giulio Chiribella, Giacomo Mauro D'Ariano, Paolo Perinotti, and Benoit Valiron. Quantum computations without definite causal structure. *Physical Review A*, 88(2):022318, aug 2013. doi:10.1103/physreva.88.022318.

18 Alexandre Clément and Simon Perdrix. PBS-Calculus: A Graphical Language for Coherent Control of Quantum Computations. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12692, doi:10.4230/LIPIcs.MFCS.2020.24.

19 J Robin B Cockett and Robert AG Seely. Proof theory for full intuitionistic linear logic, bilinear logic, and mix categories. *Theory and Applications of categories*, 3(5):85–131, 1997.

20 Timoteo Colnaghi, Giacomo Mauro D'Ariano, Stefano Facchini, and Paolo Perinotti. Quantum computation with programmable connections between gates. *Physics Letters A*, 376(45):2940–2943, October 2012. doi:10.1016/j.physleta.2012.08.028.

21 Daniel Ebler, Sina Salek, and Giulio Chiribella. Enhanced communication with the assistance of indefinite causal order. *Physical Review Letters*, 120(12):120502, March 2018. doi:10.1103/physrevlett.120.120502.

22 Adrien Feix, Mateus Araújo, and Časlav Brukner. Causally nonseparable processes admitting a causal model. *New Journal of Physics*, 18(8):083040, aug 2016. doi:10.1088/1367-2630/18/8/083040.

23 Nicolai Friis, Vedran Dunjko, Wolfgang Dür, and Hans J. Briegel. Implementing quantum control for unknown subroutines. *Physical Review A*, 89(3):030303, mar 2014. doi:10.1103/physreva.89.030303.

24 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. URL: https://www.sciencedirect.com/science/article/pii/0304397587900454, doi:10.1016/0304-3975(87)90045-4.

25 Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1, January 2007. doi:10.1145/1182613.1182614.

26 Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure v: The exponentials and splitting. *Mathematical. Structures in Comp. Sci.*, 21(3):563–584, June 2011. doi:10.1017/S096012951100003X.

27 James Hefford and Matt Wilson. A profunctorial semantics for quantum supermaps, 2024. doi:10.48550/ARXIV.2402.02997.

28 James Hefford and Matthew Wilson. A bv-category of spacetime interventions. *CoRR*, abs/2502.19022, 2025. URL: https://doi.org/10.48550/arXiv.2502.19022, arXiv:2502.19022, doi:10.48550/ARXIV.2502.19022.

29 Tony Hoare, Bernhard Möller, Georg Struth, and Ian Wehrman. Concurrent kleene algebra and its foundations. *J. Log. Algebraic Methods Program.*, 80(6):266–296, 2011. URL: https://doi.org/10.1016/j.jlap.2011.04.005, doi:10.1016/J.JLAP.2011.04.005.

30 Timothée Hoffreumon and Ognyan Oreshkov. Projective characterization of higher-order quantum transformations, 2024. URL: http://arxiv.org/abs/2206.06206, arXiv:2206.06206[quant-ph], doi:10.48550/arXiv.2206.06206.

31 Ross Horne and Alwen Tiu. Constructing weak simulations from linear implications for processes with private names. *Math. Struct. Comput. Sci.*, 29(8):1275–1308, 2019. doi:10.1017/S0960129518000452.

32 Martin Hyland and Andrea Schalk. Glueing and orthogonality for models of linear logic. *Theoretical Computer Science*, 294(1):183–231, 2003. Category Theory and Computer Science. URL: https://www.sciencedirect.com/science/article/pii/S0304397501002419, doi:10.1016/S0304-3975(01)00241-9.

33 Anna Jenčová. On the structure of higher order quantum maps, 2026. URL: https://arxiv.org/abs/2411.09256, arXiv:2411.09256.

34 Yuto Kawase. Relativized universal algebra via partial horn logic. *Theory and Applications of Categories*, 45(18):660–716, March 2026. Published 2026-03-27. URL: http://www.tac.mta.ca/tac/volumes/45/18/45-18.pdf.

35 Aleks Kissinger and Sander Uijlen. A categorical semantics for causal structure. *LMCS*, 15(3), 2019. doi:10.48550/ARXIV.1701.04732.

36 Hlér Kristjánsson, Giulio Chiribella, Sina Salek, Daniel Ebler, and Matthew Wilson. Resource theories of communication. *New Journal of Physics*, 22(7):073014, jul 2020. doi:10.1088/1367-2630/ab8ef7.

37 Hlér Kristjánsson, Tatsuki Odake, Satoshi Yoshida, Philip Taranto, Jessica Bavaresco, Marco Túlio Quintino, and Mio Murao. Exponential separation in quantum query complexity of the quantum switch with respect to simulations with standard quantum circuits, 2024. doi:10.48550/ARXIV.2409.18420.

38 Thea Li and Vladimir Zamdzhiev. Quantum coherence spaces revisited: A von neumann (co)algebraic approach. In Nathalie Bertrand and Stefan Milius, editors, *Foundations of Software Science and Computation Structures*, pages 418–439, Cham, 2026. Springer Nature Switzerland.

39 Wen-Qiang Liu, Zhe Meng, Bo-Wen Song, Jian Li, Qing-Yuan Wu, Xiao-Xiao Chen, Jin-Yang Hong, An-Ning Zhang, and Zhang-Qi Yin. Experimentally demonstrating indefinite causal order algorithms to solve the generalized deutsch's problem. *Advanced Quantum Technologies*, August 2024. doi:10.1002/qute.202400181.

40 Lê Thành Dung Nguyên and Lutz Straßburger. A system of interaction and structure III: the complexity of BV and pomset logic. *Log. Methods Comput. Sci.*, 19(4), 2023. URL: https://doi.org/10.46298/lmcs-19(4:25)2023, doi:10.46298/LMCS-19(4:25)2023.

41 Ognyan Oreshkov, Fabio Costa, and Časlav Brukner. Quantum correlations with no causal order. *Nature Communications*, 3(1), October 2012. doi:10.1038/ncomms2076.

42 Ognyan Oreshkov and Christina Giarmatzi. Causal and causally separable processes. *New Journal of Physics*, 18(9):093020, sep 2016. doi:10.1088/1367-2630/18/9/093020.

43 Erik Palmgren and Steven J. Vickers. Partial horn logic and cartesian categories. *Ann. Pure Appl. Log.*, 145(3):314–353, 2007. URL: https://doi.org/10.1016/j.apal.2006.10.001, doi:10.1016/J.APAL.2006.10.001.

44 Hugo Paquet and Philip Saville. Effectful semantics in bicategories: strong, commutative, and concurrent pseudomonads. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 61:1–61:15. ACM, 2024. doi:10.1145/3661814.3662130.

45 Lorenzo M. Procopio, Amir Moqanaki, Mateus Araújo, Fabio Costa, Irati Alonso Calafell, Emma G. Dowd, Deny R. Hamel, Lee A. Rozema, Časlav Brukner, and Philip Walther. Experimental superposition of orders of quantum gates. *Nature Communications*, 6(1), aug 2015. doi:10.1038/ncomms8913.

46 Tom Purves and Anthony J. Short. Quantum theory cannot violate a causal inequality. *Phys. Rev. Lett.*, 127:110402, Sep 2021. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.110402, doi:10.1103/PhysRevLett.127.110402.

47 Christian Retoré. Pomset logic: A non-commutative extension of classical linear logic. In Philippe de Groote, editor, *Typed Lambda Calculi and Applications, Third International Conference on Typed Lambda Calculi and Applications, TLCA '97, Nancy, France, April 2-4, 1997, Proceedings*, volume 1210 of *Lecture Notes in Computer Science*, pages 300–318. Springer, 1997. doi:10.1007/3-540-62688-3_43.

48 Christian Retoré. Pomset logic: a logical and grammatical alternative to the lambek calculus. *CoRR*, abs/2001.02155, 2020. URL: http://arxiv.org/abs/2001.02155, arXiv:2001.02155.

49 Peter Selinger. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 14(4):527–586, aug 2004. doi:10.1017/s0960129504004256.

50 Peter Selinger. Towards a semantics for higher-order quantum computation. In *Proceedings of the 2nd International Workshop on Quantum Programming Languages*, pages 127–143, 2004.

51 Will Simmons and Aleks Kissinger. Higher-order causal theories are models of bv-logic, 2022. doi:10.48550/ARXIV.2205.11219.

52 Will Simmons and Aleks Kissinger. A complete logic for causal consistency. *CoRR*, abs/2403.09297, 2024. URL: https://doi.org/10.48550/arXiv.2403.09297, arXiv:2403.09297, doi:10.48550/ARXIV.2403.09297.

53 Márcio M. Taddei, Jaime Cariñe, Daniel Martínez, Tania García, Nayda Guerrero, Alastair A. Abbott, Mateus Araújo, Cyril Branciard, Esteban S. Gómez, Stephen P. Walborn, Leandro Aolita, and Gustavo Lima. Computational advantage from the quantum superposition of multiple temporal orders of photonic gates. *PRX Quantum*, 2(1):010320, feb 2021. doi:10.1103/prxquantum.2.010320.

54 Alwen Tiu. A system of interaction and structure II: the need for deep inference. *Log. Methods Comput. Sci.*, 2(2), 2006. doi:10.2168/LMCS-2(2:4)2006.

55 Julian Wechs, Hippolyte Dourdent, Alastair A. Abbott, and Cyril Branciard. Quantum circuits with classical versus quantum control of causal order. *PRX Quantum*, 2(3):030335, August 2021. doi:10.1103/prxquantum.2.030335.

56 Wataru Yokojima, Marco Túlio Quintino, Akihito Soeda, and Mio Murao. Consequences of preserving reversibility in quantum superchannels. *Quantum*, 5:441, April 2021. doi:10.22331/q-2021-04-26-441.

57 Gaoyan Zhu, Yuanbo Chen, Yoshihiko Hasegawa, and Peng Xue. Charging quantum batteries via indefinite causal order: Theory and experiment. *Phys. Rev. Lett.*, 131:240401, Dec 2023. URL: https://link.aps.org/doi/10.1103/PhysRevLett.131.240401, doi:10.1103/PhysRevLett.131.240401.