`QS`: Quantum Programming via Linear Homotopy Types

David J. Myers[*] Hisham Sati[*†] Urs Schreiber[*]

August 24, 2023

Abstract

We lay out a language paradigm, `QS`, for quantum programming and quantum information theory – rooted in the algebraic topology of stable homotopy types – which has the following properties, deemed necessary and probably sufficient for the eventual goal of heavy-duty quantum computation:

- **Application**: in its 0-sector, `QS` is cross-translatable with the established quantum programming scheme `Quipper`, including support for classical control (dynamic lifting via dependent linear types) such as by quantum measurement outcomes which are handled monadically as in the widely used `zxCalculus`.
- **Compilation**: but `QS` is embedded in (is just syntactic sugar for) a universal quantum certification language `LHoTT`, being a novel linear enhancement of the established formal (programming/certification) language scheme of Homotopy Type Theory (`HoTT`).
- **Certification**: as such, `QS` introduces a previously missing method of formal verification of general classically controlled quantum programs, e.g. it verifies quantum axioms such as the deferred measurement principle.
- **Stabilization**: in its higher sector, `QS` natively models hardware-level topologically stabilized quantum computation such as by realistic anyonic braid gates, verifying their conformal field theoretic properties.
- **Realization**: in fact, `QS` naturally interfaces with the holographic quantum theory of topologically ordered quantum materials that are thought to eventually provide topologically stabilized quantum hardware.

In developing these results we find a pleasant unification of *quantum logic* (linear types), *epistemic modal logic* (possible worlds), *quantum interpretations* (many worlds), and *twisted cohomology* (parameterized spectra) & *motives* (six-operations) – which may be of interest in itself. ("`QS`" stands both for "Quantum Systems language" and for the sphere spectrum "QS^0".)

In companion articles [TQP][EoS], we further discuss topological quantum gates in and the categorical semantics of `LHoTT`/`QS`.

Contents

---

[*] Mathematics, Division of Science; and
Center for Quantum and Topological Systems,
NYUAD Research Institute,
New York University Abu Dhabi, UAE.

CENTER FOR
QUANTUM &
TOPOLOGICAL
SYSTEMS

[†] The Courant Institute for Mathematical Sciences, NYU, NY

The authors acknowledge the support by *Tamkeen* under the *NYU Abu Dhabi Research Institute grant* `CG008`.

incomplete draft
in preparation

for the latest version and related material see:

ncatlab.org/schreiber/show/Quantum+Certification+via+Linear+Homotopy+Types

1 Introduction

We lay out an approach to a joint solution of the following open problems:

(I) The open problem of reliable quantum computing. While the hopes associated with quantum computing (Lit. 2.1) are hard to overstate, experts are well-aware [1] that currently existing hard- and soft-ware paradigms are unlikely to support the desired heavy-duty quantum computations beyond toy examples. The two fundamental open problems that the field still faces are both rooted in the single most enigmatic and proverbial phenomenon of quantum physics: the *state collapse* or *decoherence* phenomenon (Lit. 2.2), whereby the peculiar non-classical properties of quantum systems on which rest the hopes of quantum computing are jeopardized by any measurement-like interaction of the system's environment. This means that scalably robust quantum computing requires:

- **(i) Topological hardware** (Lit. 2.3) given by quantum materials whose registry-states are protected by an "energy gap" from having *any* interaction with the environment below that range.
- **(ii) Verified software** (Lit. 2.4) with compile-time certificates of correctness, since the traditional run-time debugging of complex programs is impossible for quantum programs (causing collapse), while all the more needed due to the complexity and intransparency of gate-level quantum circuits.

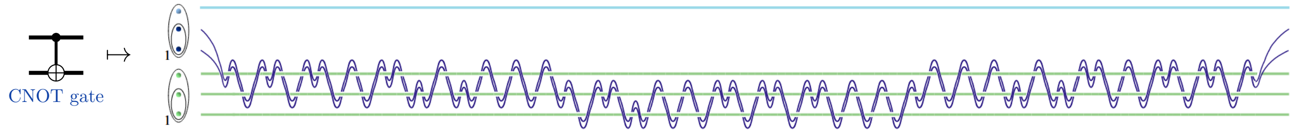
Both of these issues have been discussed separately, but the necessary combination has remained essentially untouched until [TQP]; one will need a quantum programing language (Lit. 2.5) which is

- **(iii) certifiable and topological-hardware-aware**, allowing the programmer to formally verify at compile-time the correctness not (just) of high-level quantum programs, but of quantum circuits consisting of the peculiar topological quantum gates that the topological quantum hardware actually provides.

For example, to state just the most immediate problem:

> **Topological quantum circuit compilation problem** (Lit. 2.7).
> *Suppose a topologically ordered quantum material is finally developed which features* su_2*-anyon states at level* ℓ*, and given any quantum circuit written in the usual QBit-basis, then the quantum compilation of this circuit onto the given hardware is the specification of a braid (an element of a braid group) such that the holonomy of the* su_2^ℓ *Knizhnik-Zamolodchikov connection along the corresponding path in the configuration space of defect points in the given quantum material may be conjugated onto the unitary operator to which the quantum circuit evaluates, within a specified accuracy.*

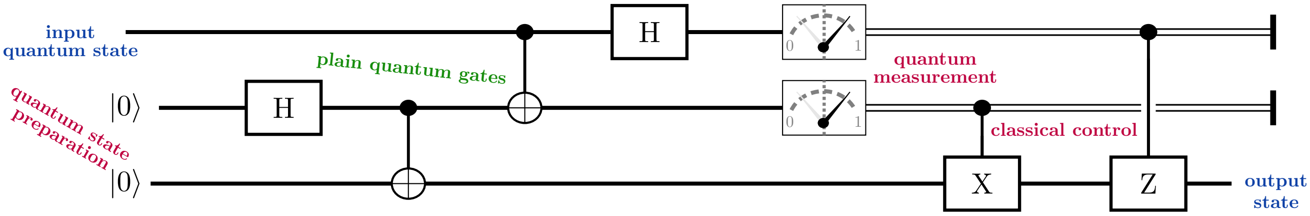
Here the relevant braids are humongous while having no recognizable resemblance to the quantum algorithm which they are executing; for instance, a single CNOT gate (8) may compile to the following braid [HZBS07, Fig. 15]:



Hence future quantum programmers will anyways need (classical) computer assistance to compile their quantum programs onto topological hardware. To make that intricate process fail-safe to reliably run on precious scarce quantum resources, we need this computer algebra to be "aware" of the system specification and to certify its own correctness relative to this specification.

And this is just for the simplest case of no classical control. The general problem is harder still:

The problem of certifying classical control. Even the most elementary quantum information protocols involve mid-circuit measurement and classical control, such as in the quantum teleportation protocol (cf. §6.2):



---

[1] [Sau17]: "small machines are unlikely to uncover truly macroscopic quantum phenomena, which have no classical analogs. This will likely require a scalable approach to quantum computation. [...] based on [...] topological quantum computation (TQC) as envisioned by Alexei Kitaev and Michael Freedman [...] The central idea of TQC is to encode qubits into states of topological phases of matter. Qubits encoded in such states are expected to be topologically protected, or robust, against the 'prying eyes' of the environment, which are believed to be the bane of conventional quantum computation."

[DS22]: "The qubit systems we have today are a tremendous scientific achievement, but they take us no closer to having a quantum computer that can solve a problem that anybody cares about. [...] What is missing is the breakthrough [...] bypassing quantum error correction by using far-more-stable qubits, in an approach called topological quantum computing."

More importantly, beyond the currently available NISQ paradigm (Lit. 2.8), serious quantum computation is expected (Lit. 2.9) to involve a perpetual loop of classical control operations on the quantum computer (*hybrid* classical/quantum computation). These are predominantly for quantum error correction (§6.3) but also for purposes such as repeat-until-success gates (§6.4) – where subsequent quantum circuit execution is classically conditioned on run-time quantum measurement results – also called "dynamic lifting" (Lit. 2.9, namely of quantum measurement results into the classical data register). This is schematically indicated on the right.

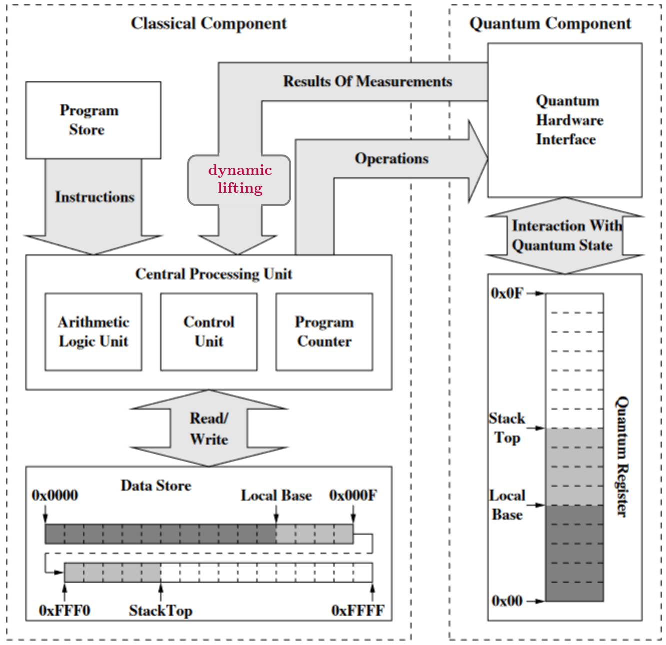
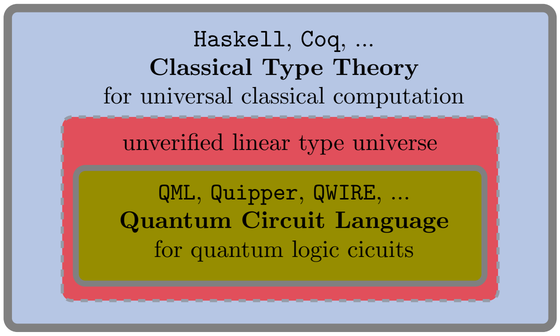


diagram adapted from [NPW07, Fig. 1]

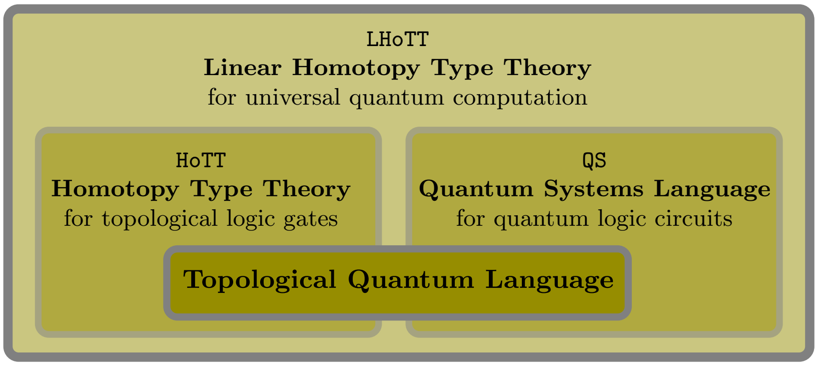
Hence what is needed for reliable quantum computation is a certification language that knows about classical data types *and* about linear/quantum data types *and* their *dependency* on classical data. This had been lacking:

The problem of embedded quantum languages. Namely, for previous lack of a *universal* quantum programming language, existing quantum circuit languages are embedded into *classical* host languages ([RS20][GLRSV13] [RPZ18][PZ19][HRHWH21][HRHLH21][ZBSLY23]) which do not have native support for linear types (cf. Lit. 2.4) nor for classical control of quantum circuits. For instance, basic protocol schemes such as quantum teleportation (§6.2), quantum error correction (§6.3) or repeat-until-success gates (§6.4) remain unverifiable with previous technology.

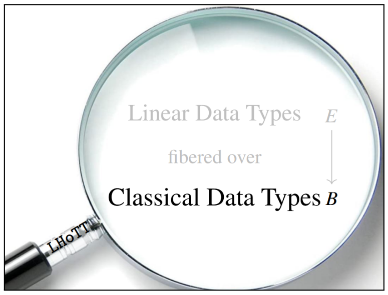


Solution by Linear Homotopy Type Theory. We argue here, as announced in [Sch22], that the novel type theory `LHoTT` (§3) recently developed in [Ri22] (as anticipated in [Sch14b]) in extension of the classical language scheme `HoTT` (Lit. 2.6) serves as the missing universal quantum programming/certification language. Our claim is that `LHoTT`:

- Solves the old problem of constructing combined classical/linear type theories (cf. Lit. 2.4).
- Provides existing quantum programming languages like `Quipper` with a certification mechanism [Ri23].
- Natively supports quantum effects such as dynamic lifting of run-time quantum measurement (§4).
- Natively supports verification of realistic topological quantum gates [TQP].



We argue that this makes `LHoTT`/`QS` the first comprehensive paradigm for serious quantum programming beyond the NISQ area; see §7 for outlook.



Concretely, `LHoTT` enhances the syntactic rules of classical `HoTT` by further type formations which serve to exhibit every (homotopy) type E of the language as secretly consisting of an underlying classical (intuitionistic) base type $B \equiv \natural E$ equipped, in a precise sense, with a microscopic (infinitesimal) halo of linear/quantum data. As such, `LHoTT` may neatly be thought of as the formal logical expression of a microscope that resolves quantum aspects on structures that macroscopically appear classical. This way `LHoTT` embeds quantum logic into classical logic in a way reminiscent of Bohr's famous dictum[2]that all quantum phenomena must be expressible in classical language.

Formally this is achieved by, first of all, adjoining to classical `HoTT` an *ambidextrous* modal operator $\natural$ [RFL21] (an *infinitesimal cohesive modality* [Sch13, Def. 3.4.12, Prop. 4.1.9]), whose modal types (Lit. 2.14) are the *purely classical* (ordinary) homotopy types, embedded *bi-reflectively* (92) among all data types (more in §3):

The presence of the $\natural$-modality exhibits general types $E : \mathrm{Type}$ as microscopic/infinitesimal *halos* around their underlying purely classical type $\natural E : \mathrm{ClaType}$. It is a profound fact (17) of ∞-topos theory that models for such *infinitesimal cohesion* (see Lit. 2.10) are provided by parameterized module spectra, in particular by flat ∞-vector bundles ("∞-local systems", see [EoS]) which, in their 0-sector (Rem. 2.11), accommodate quantum circuit semantics (cf. §4.3) in indexed sets of vector spaces (cf. §3.1) such as known from the Proto-`Quipper` quantum language (Lit. 2.5).

$$\begin{tikzcd}
 & \text{classical modality} & & \text{flat } \infty\text{-vector bundles} \\
 & & & (\infty\text{-local systems}) \\
\text{bundles of linear homotopy types} & \mathrm{Type} \arrow[loop above, "\natural"] \arrow[d, shift right=4, "p"'] \arrow[d, shift left=4, "p"] \arrow[d, phantom, "\dashv\, 0 \,\dashv"] \arrow[d, shift right=8, phantom, "\text{bireflection}"'] & \overset{\text{e.g.}}{=} & \int_{\mathbf{X}} \mathbf{sCh}^{\mathbf{X}}_{\mathbb{K}} \arrow[d, shift right=2, "\text{base space}"'] \arrow[d, phantom, "\dashv"] \\
\text{purely classical homotopy types} & \mathrm{ClaType} \arrow[u] & \overset{\text{e.g.}}{=} & \{\mathbf{X} \in \mathrm{sSet\text{-}Grpd}\} \arrow[u, shift right=2, "\text{zero-section}"']
\end{tikzcd} \tag{1}$$

Linear homotopy theory as the organizing principle. Generally, our thesis (following [Sch14a][Sch14b][IHH]) is that the conceptual foundation not just of quantum computing but in fact of fundamental quantum physics generally is in *linear homotopy theory*, by which we refer to what is alternatively known (Lit. 2.10):

- in algebraic topology as the *indexed ∞-category* of *parameterized module spectra* (cf. [EoS, Rem. 3.4.1]),
- in algebraic geometry essentially as the *yoga of six operations* on *motives* (cf. [EoS, pp. 41]),
- in higher topos theory as the theory of *tangent ∞-toposes* or *Joyal loci*,
- in cohomology theory as the subject of *twisted generalized cohomology theory* with its base change operations.

In the following we incrementally unwind what this means and how it relates to quantum systems and serves to express quantum programming with topological effects.

(...)

$H\mathbb{C}$**-Linear quantum theory.** In this scheme, conventional quantum information theory happens in the $\mathbb{C}$-linear form of linear homotopy theory (see [EoS]) where parameterized $H\mathbb{C}$-module spectra are equivalent to *flat ∞-bundles of chain complexes*, also known as *∞-local systems.* Here the higher structure of chain complexes serves to capture topological quantum effects [TQP], but in the 0-truncated sector these are just set-indexed complex vector spaces of the form familiar from the categorical semantics of the quantum language `Quipper`; and much of our discussion below focuses on laying out the structures in this 0-truncated $\mathbb{C}$-linear sector in much detail, showing that it is a streamlined and convenient context for traditional quantum information theory and for quantum computing with classical control. At the same time, since all structures (such as quantum measurement effects) are encoded modally/monadically, this discussion straightforwardly generalizes away from the $\mathbb{C}$-linear 0-truncated sector.

(...)

KR**-Linear quantum theory.** However, besides the higher topological generalization it provides, linear homotopy theory also exists beyond the $\mathbb{C}$-linear sector, encompassing homotopy-theoretic enhancements of linear algebra once known as *brave new algebra*, where the ground ring $\mathbb{C}$ is replaced by a *ring spectrum* representing a multiplicative generalized cohomology theory. We had already argued in [SS23b] that the precise description of topologically ordered phases of quantum materials requires linearity over the equivariant ring spectrum KR, but here (in §5) we explain that, even more fundamentally, the probabilistic content of quantum theory *emerges* in KR-linear homotopy theory. Practically this means, as we will explain, that our language `LHoTT` natively supports quantum circuits not just of pure but also of mixed states (density matrices).

(...)

---

[2] [Bohr1949, pp. 209]: "however far the phenomena transcend the scope of classical physical explanation, the account of all evidence must be expressed in classical terms". For background and commentary see also [Sche73, p. 24].

(II) The open problem of formalizing quantum epistemic logic. With the need for a universal and verifiable quantum programming language established, the next open problem is that of language design, which here we mean in a fundamental paradigmatic way:

Given that dependent type theory is the fundamental paradigm for certified programming in general (Lit. 2.4), *what makes it applicable to certification of quantum effects such as quantum measurement* (Lit. 2.2)?

Notice here that a universal quantum programming language has to accurately reflect the logical content of quantum physics, where the act of formulating a quantum program is as well that of recounting, in formalized language, the physical process of its execution, *including* processes of quantum measurement and hence including the curious nature of quantum epistemology. In this sense, we may claim that:

Finding a universal quantum programming language means finding a formal language for quantum epistemology.

The role of modal logic. Stated this way, we need not look much further for guidance on the matter, since the formal language paradigm for dealing with questions of epistemology has long been understood to be *modal logic* (Lit. 2.13), where the usual logical connectives are accompanied by formal expressions for qualified *modes* in which propositions may hold, such as *necessarily* ($\Box$) or *possibly* ($\Diamond$) namely (which is the perspective of relevance here:) for all or any *measurement outcome* that may be obtained, or *possible world* w (as the modal logician says) that one may find oneself in, one of the *many worlds* (as the quantum philosopher says):

$$
\overset{\substack{\textbf{Set of many possible worlds}\\\textbf{(of measurement outcomes)}}}{W : \mathrm{Set}\,,}
\qquad
\overset{\substack{W\textbf{-dependent}\\\textbf{proposition}}}{P \,:\, \mathrm{Prop}_W}
\qquad
\overset{\text{yields that}}{\vdash}
\qquad
\left.
\begin{array}{c}
\overset{\substack{\textbf{“}P\textbf{ holds necessarily”}\\\text{(no matter the outcome/world)}}}{\Box P \;\equiv\; \forall_w\, P(w)}\\[6pt]
\underset{\substack{\textbf{“}P\textbf{ holds possibly”}\\\text{(for some outcome/world)}}}{\Diamond P \;\equiv\; \exists_w\, P(w)}
\end{array}
\right\}
\;\overset{\text{is a}}{:}\;
\overset{\substack{W\textbf{-independent}\\\textbf{proposition}}}{\mathrm{Prop} \hookrightarrow \mathrm{Prop}_W}
\tag{2}
$$

If here we think of classical propositions as certain data types (namely of data that certifies their assertion), then it is natural to generalize this from modal logic to *modal type theory* (Lit. 2.14) where we consider any W-dependent data types:[3]

$$
\overset{\substack{\textbf{Type of many possible worlds}\\\textbf{(of measurement outcomes)}}}{W : \mathrm{Type}\,,}
\qquad
\overset{\substack{W\textbf{-dependent}\\\textbf{data type}}}{D \,:\, \mathrm{Type}_W}
\qquad
\overset{\text{yields that}}{\vdash}
\qquad
\left.
\begin{array}{c}
\overset{\substack{\textbf{type of }D\textbf{-data for}\\\textbf{every world/outcome}}}{\Box D \;\equiv\; \prod_w\, D(w)}\\[6pt]
\underset{\substack{\textbf{type of }D\textbf{-data}\\\textbf{for any world/outcome}}}{\Diamond D \;\equiv\; \coprod_w\, D(w)}
\end{array}
\right\}
\;\overset{\text{is a}}{:}\;
\overset{\substack{W\textbf{-independent}\\\textbf{data type}}}{\mathrm{Type} \hookrightarrow \mathrm{Type}_W}
\tag{3}
$$

Epistemic modal logic as Dependent type theory. Remarkably, in this more general form (3) the system *simplifies* since this *epistemic modal type theory* is just plain dependent type theory with the W-dependent type formation rules viewed not as adjoints but equivalently as (co)*monadic* modalities (Lit. 2.15, 2.14):

We observe in §4.1 that possible-world semantics for modal logic (in its "S5" flavor with which we are concerned here) is equivalently that induced by dependent type formation along any context extension. Conversely, this means to observe (Rem. 4.1) that one may think of standard dependent type theory as epistemic modal type theory with a universal system of epistemic modal operators indexed by types of "many possible worlds" W : Type. From this perspective, the tradition in formal logic to refer to the large type Type of small types as the "universe" gains some vindication.

$$
\begin{tikzcd}[column sep=huge, row sep=large]
\mathrm{Type}_W
  \arrow[loop above, "\overset{\textbf{possibility modality}}{\Diamond_W}"]
  \arrow[loop below, "\underset{\textbf{necessity modality}}{\Box_W}"']
  \arrow[r, shift left=3, "\overset{\textbf{dependent ``sum''}}{\coprod_W}"]
  \arrow[r, shift right=3, "\underset{\textbf{dependent product}}{\prod_W}"']
& \mathrm{Type}
  \arrow[l, "\times W" description]
  \arrow[l, phantom, shift right=1.5, "\perp"]
  \arrow[l, phantom, shift left=1.5, "\perp"]
\end{tikzcd}
\tag{4}
$$

While for classical intuitionistic type theory, this perspective may be of interest to the analytic philosopher (see [Co20, Ch. 4]), we next claim that applied to *linear* dependent type theory the same perspective solves the practical problem of formalized quantum epistemology relevant for universal quantum programming/certification:

---

[3] We write "$\coprod_w$" for the (non-linear) type formation traditionally referred to as "dependent sum" and traditionally denoted "$\sum_w$", since the latter symbol is borrowed from linear algebra, an (unnecessary) abuse of notation that becomes untenable after our passage from classical intuitionistic to actual linear dependent type theory.

Quantum epistemic logic as Linear dependent type theory. The point is that in linear dependent type theory `LHoTT` the situation (4) has an immediate analog ([Ri22, §2.4]) as W-dependent classical intuitionistic types are replaced by W-dependent *linear* types (quantum data types): In this case and assuming W is *finite* (as it is for any realistic quantum measurement) their linear/quantum nature makes the dependent (co)product adjoints coincide ("ambidexterity", Lit. 2.16) on the *direct sum* of linear types, this reflecting the superposition principle of quantum physics:

linear possibility **direct sum** **linear randomness**

Frobenius monad of quantum epistemic logic (§4.2) proves principles as *deferred measurement* (Prop. 4.16)

$$\begin{tikzcd}
W : \mathrm{FinClaType} \;\vdash & \mathrm{QuType}_W \arrow[loop above, "\Diamond_W"] \arrow[loop below, "\Box_W"'] \arrow[r, shift left=4, "\oplus_W"] \arrow[r, shift right=4, "\oplus_W"'] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathrm{QuType} \arrow[l] \arrow[loop above, "\star_W"] \arrow[loop below, "\bigcirc_W"']
\end{tikzcd}$$

Classical context ?? Frobenius monad as in `zxCalculus` gives *effect-logic* for quantum gates §4.3

linear necessity **direct sum** **linear indefiniteness**

This means equivalently that in the linear case the (co)monadic modal operators coincide, $\Diamond_W \simeq \Box_W\,, \ \star_W \simeq \bigcirc_W$ (5) to form a pair of *Frobenius monads* (cf. Prop. 4.14), reflecting the monadic nature of quantum measurement as known from the `zxCalculus` (Lit. 2.16). It may be satisfactory to observe that the modal-logical expression of this situation reflects Gell-Mann's *principle of quantum compulsion* (cf: [Bu76, p. 31]: "In quantum physics anything that is not forbidden [i.e, possible] is compulsory [i.e., necessary]."):

Finite classical type of many possible worlds (measurement outcomes) — **W-dependent quantum data type** — yields that — linear sum — is a — **W-independent quantum data type**

$$\begin{tikzcd}
 & & & & \underset{w}{\oplus} D_w \arrow[dr, "\sim"] & & & \\
W : \mathrm{FinClaType}\,, & D \,:\, \mathrm{QuType}_W & \vdash & \Diamond D \arrow[ur, "\sim"] \arrow[rr, "\sim"'] & & \Box D & : & \mathrm{QuType} \hookrightarrow \mathrm{QuType}_W
\end{tikzcd} \tag{6}$$

the possible is necessary
principle of quantum compulsion

We suggest thinking of this as a Yoneda-Lemma-type statement: The derivation of (5) is so elementary that it borders on being tautological, and yet as an organizing principle for quantum effects we will find it to be ubiquitous, for instance in implying the *deferred measurement principle* (Prop. 4.16) or the commuting diagram (7) below, which arguably makes precise many words [Te98] written in the informal literature on the matter. This leads one to wonder: Had history proceeded differently, could systematic development of combined modal and linear logic have led pure logicians to discover the rules of quantum information theory independently of experimental input?

Formal logic of quantum measurement effects. Remarkably, unwinding the logical rules of this epistemic quantum logic (6) reveals that it knows all about the state collapse after quantum measurement including formal proof of its equivalence to *branching* into "many worlds" (Lit. 2.2):

The hexagon of quantum epistemic entailments. A commuting diagram (130) of implications in the quantum modal logic (4) for the case of a QBit-measurement-controlled quantum gate $G_\bullet$ on a quantum register of the form $\mathcal{H} \equiv \Box_{\mathrm{Bit}}\mathcal{H}_\bullet = \mathcal{H}_0 \oplus \mathcal{H}_1$ (e.g. $\mathcal{H} = H \otimes \mathrm{QBit}$ if $\mathcal{H}_i = H$).

Classical register — **quantum register** — QBit-**measurement branching** (pp. 70) — QBit-**controlled quantum gate** — **dynamic lifting**

$$\begin{tikzcd}
 & & \mathcal{H} \arrow[rr, "G_0"] & & \mathcal{K} \arrow[dr, "\delta_0^{b}"] & \\
b : \mathrm{Bit} \;\vdash & \mathcal{H} \arrow[ur, "P_0"] \arrow[dr, "P_1"'] & \oplus & \oplus & \oplus & \mathcal{K} \\
 & & \mathcal{H} \arrow[rr, "G_1"'] & & \mathcal{K} \arrow[ur, "\delta_1^{b}"'] & \\
 & & \Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}\mathcal{H}_\bullet \arrow[rr, "\Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}G_\bullet"] & & \Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet \arrow[dr, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet}"] & \\
 & \Box_{\mathrm{Bit}}\mathcal{H}_\bullet \arrow[ur, "\Box_{\mathrm{Bit}}\left(\mathrm{ret}^{\Diamond_{\mathrm{Bit}}}_{\mathcal{H}_\bullet}\right)"] \arrow[dr, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathcal{H}_\bullet}"'] \arrow[rrrr, "\boxed{\text{classically controlled quantum computing cycle}}" description] & & & & \Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet \\
 & & \mathcal{H}_\bullet \arrow[rr, "G_\bullet"] & & \mathcal{K}_\bullet \arrow[ur, "\mathrm{ret}^{\Diamond_{\mathrm{Bit}}}_{\mathcal{K}_\bullet}"'] & \\
b : \mathrm{Bit} \;\vdash & \mathcal{H} \arrow[r] & \mathcal{H}_{b} \arrow[rr, "G_{b}"] & & \mathcal{K}_{b} \arrow[r, hook] & \mathcal{K} \\
 & \sum\limits_{b' \in \mathrm{Bit}} \vert \psi_{b'} \rangle \arrow[r, mapsto] & \vert \psi_{b} \rangle \arrow[rr, mapsto] & & G_{b} \vert \psi_{b} \rangle &
\end{tikzcd} \tag{7}$$

quantum effects Everett-style

quantum effects Copenhagen-style

QBit-**measurement collapse** (pp. 67) — **quantum gate conditioned on classical control logic** — **dynamic** QBit-**state preparation**

Moreover, the (co)monadic formalization of quantum measurement in the `zxCalculus` (Lit. 2.16) derives from this formulation (cf. Prop. 4.14, Rem. 4.18). Using standard translation (Lit. 2.15) of such (co)monadic effects into programming language constructs yields (in §6) a quantum certification language `QS` embedded in `LHoTT`.

(III) The open problem of strongly-correlated quantum materials. Interestingly, these fundamental theoretical problems at the foundations of quantum computing closely relate to a glaring open problem in condensed matter and quantum field theory (Lit. ...):

Namely in asking for topological-hardware-aware quantum languages, we are effectively asking for a formal language of topologically ordered phases of matter (Lit. ...). But since these are *strongly interacting* (*strongly-correlated*) quantum systems, they tend to fall outside the established scope of what traditional *perturbative* methods of quantum field theory apply to. The problem of understanding *non-perturbative* strongly-coupled quantum systems is a general one, which is might be most famous for its guise of the *confinement problem* in elementary particle physics, where it has been pronounced a mathematical *Millennium Problem* by the Clay Mathematics Institute.

It may seem overambitious that in a treatise on quantum programming, we should have anything to say about problems in quantum field theory, but we offer the inclined reader an argument (exposition in [IHH], more details in [SS23b, Rem. 2.6][SS23a]) that the solutions to these fundamental problems share a common root in *linear homotopy theory* (in the sense of p. 5) and as such lend themselves to formulation in `LHoTT`.

(...)

Certainly no background from [IHH] is assumed here, but the reader familiar with this angle of fundamental physics may understand the present discussion as comprehensively expanding on the general picture of quantum theory used there.

(...)

Outline:

§3 on the linear type system and its formalization in `LHoTT`,
§4 on the induced monadic quantum (measurement) effects,
§5 on generalization to mixed states and quantum probability,
§6 on pseudocode for an embedded quantum language `QS`.

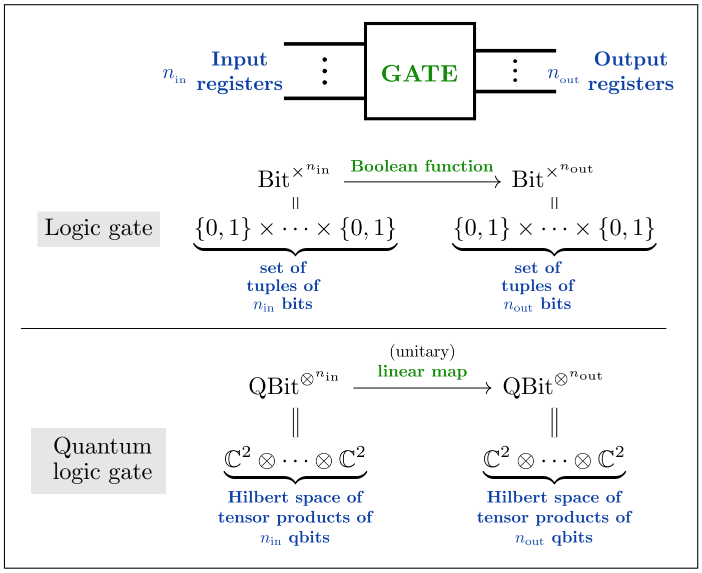
2 Background

This section provides background information and pointers to the literature on various subjects referred to in the main text. All items here are separately well-known to their respective experts but not always easy to comprehensively glean from the literature. We pause at times to point out the remaining gaps that we address in the main text. The reader may or may not want to read this section linearly; we will refer back to here as the need arises.

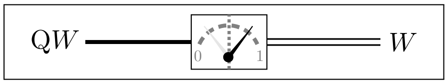
Literature 2.1 (**Quantum computation and Quantum information processing**).
The basic idea of *quantum computation* and *quantum information processing* is to exploit, for the purpose of machine computation and information processing, the peculiar laws of quantum physics (Lit. 2.2) – which are obeyed by *undisturbed* (Lit. 2.3) microscopic systems.

The general idea of quantum computation was originally articulated by Yuri Manin [Ma80][Ma00], Paul Benioff [Be80], and Richard Feynman [Fey82][Fey86], brought into shape by David Deutsch [De89], shown to be potentially of dramatic practical relevance by Peter Shor and others [Sh94][Si97]... *if* sufficient quantum coherence could be technologically retained (cf. Lit. 2.3), which has so far been achieved only marginally (Lit. 2.8).

Textbook accounts of the general principles of quantum computation and quantum information theory include: [NC10][RP11][BCR18][BEZ20]. Impressions of the state of the field may be found in [Pr22]. An exposition leading up to the following discussion may be found in [Sch22].



The idea of quantum gates. It is a standard concept in computer science to speak of *logic gates* (e.g. [GMSW21, §1]) for operations on classical memory/registers (typically but not necessarily on a set of "bits", hence of Boolean "truth values", whence the name) – where the terminology suggests but need not imply that this is an *elementary* operation performed by some computing machine under consideration. The evident analog in quantum computation (Lit. 2.1) is that of *quantum logic gates* ([Fey86][De89][BBCDMSSSW95], often called just "quantum gates", for short) which are *linear* maps acting on some quantum memory/registers – typically imagined to be constituted by "qbits" (66). In classically controlled quantum computation (Lit. 2.9) one is dealing with *classically controlled quantum gates* (e.g. [NC10, §4.3]) that read/write a combination of classical and quantum data.
A basic example of a (controlled, quantum) logic gate is the *controlled NOT gate* [De89, Fig. 2] (`CNOT` for short, cf. [NC10, §1.3.2]) which operates on a pair of (q)bits by inverting the second conditioned on the first; see figure (8).



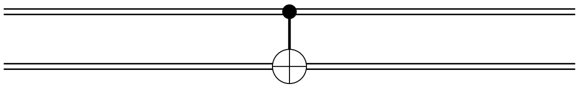
Quantum measurement gates. One also wants to regard the operation of *quantum measurement* itself (Lit. 2.2) as a quantum gate (e.g. [NC10, p. xxv]), whose input is quantum data but whose output is the classical measurement result.

Notice that the proper data-typing (Lit. 2.4) of a quantum measurement gate is more subtle than that of an ordinary logic gate, since the actual measurement outcome is *not* determined by the gate's input data (and hence *not* knowable at "compile time" of a quantum program) but is a fundamentally indefinite result, more akin to operations otherwise considered in the field of (classical but) *nondeterministic* computation (e.g. [Sip12, §1.2]).

Beware that this is not a side issue but part of the crux of quantum computation: On the one hand, the stochastic nature of quantum measurement is a *fundamental* principle of physics (certainly of presently accessible physics, see Lit. 2.2) and not just a reflection of incomplete knowledge about a quantum system (in contrast to, for instance, the case of classical thermodynamics). Moreover, state collapse under quantum measurement is not just a subjective update of expected probabilities, in that it objectively serves as an operational logic gate in quantum computations (such as in quantum teleportation §6.2 and quantum error correction §6.3), to the extent that any quantum computation may be realized by *exclusively* using (quantum state preparation and) quantum measurement gates (known as "measurement-based quantum computation"; cf. [Nie03][BBDRV09][Wei21]).

We discover a natural way for dealing with formal typing of quantum measurement below in §4.3.

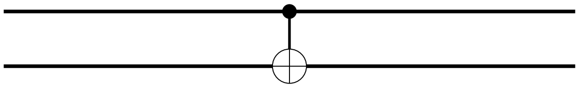
`CNOT` logic gates



Controlled NOT gate
(reversible XOR gate)

$$\mathrm{Bit}^2 \xrightarrow[= \left(b_1,\ b_1 \operatorname{xor} b_2\right)]{(b_1, b_2) \ \longmapsto\ \left(b_1,\ \begin{smallmatrix}\text{if } b_1 \text{ then } \neg b_2 \\ \text{else } \ b_2\end{smallmatrix}\right)} \mathrm{Bit}^2$$

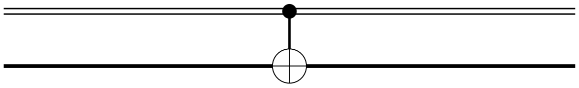
$$\{0,1\} \times \{0,1\} \xrightarrow{\begin{smallmatrix}(0,0) & \mapsto & (0,0) \\ (0,1) & \mapsto & (0,1) \\ (1,0) & \mapsto & (1,1) \\ (1,1) & \mapsto & (1,0)\end{smallmatrix}} \{0,1\} \times \{0,1\}$$



Quantumly controlled
quantum NOT gate

$$\mathrm{QBit}^{\otimes^2} \xrightarrow{\sum\limits_{b_1,b_2} q_{b_1,b_2} |b_1\rangle \otimes |b_2\rangle \ \longmapsto\ \sum\limits_{b_1,b_2} q_{b_1,b_2} |b_1\rangle \otimes |b_1 \operatorname{xor} b_2\rangle} \mathrm{QBit}^{\otimes^2} \tag{8}$$

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \xrightarrow{\begin{smallmatrix}|0\rangle\otimes|0\rangle & \mapsto & |0\rangle\otimes|0\rangle \\ |0\rangle\otimes|1\rangle & \mapsto & |0\rangle\otimes|1\rangle \\ |1\rangle\otimes|0\rangle & \mapsto & |1\rangle\otimes|1\rangle \\ |1\rangle\otimes|1\rangle & \mapsto & |1\rangle\otimes|0\rangle\end{smallmatrix}} \mathbb{C}^2 \otimes \mathbb{C}^2$$

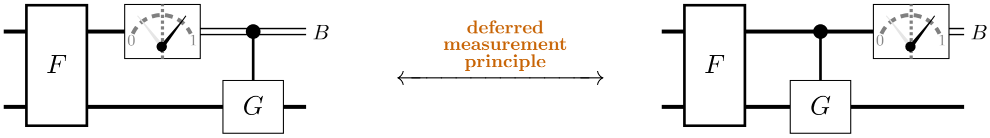


Classically controlled
quantum NOT gate

$$\mathrm{Bit} \times \mathrm{QBit} \xrightarrow{\left(b_1, \sum\limits_{b_2} q_{b_2} |b_2\rangle\right) \ \longmapsto\ \left(b_1, \sum\limits_{b_2} q_{b_2} |b_1 \mathrm{xor} b_2\rangle\right)} \mathrm{Bit} \times \mathrm{QBit}$$

$$\{0,1\} \times \mathbb{C}^2 \xrightarrow{\begin{smallmatrix}(0,|0\rangle) & \mapsto & (0,|0\rangle) \\ (0,|1\rangle) & \mapsto & (0,|1\rangle) \\ (1,|0\rangle) & \mapsto & (1,|1\rangle) \\ (1,|1\rangle) & \mapsto & (1,|0\rangle)\end{smallmatrix}} \{0,1\} \times \mathbb{C}^2$$

Deferred measurement principle. Since quantum measurement turns quantum data into classical data, it intertwines quantum control with classical control. Concretely, a statement known as the *deferred measurement principle* asserts that any quantum circuit containing intermediate (mid-circuit) quantum measurement gates followed by gates conditioned on the measurement outcome is equivalent to a circuit where all measurements are "deferred" to the last step of the computation



(9)

(In the practice of quantum computation this principle can be used to optimize quantum circuit design. More philosophically, it is interesting to notice that the issue of epistemological puzzlement in quantum interpretations, Lit. 2.2, can always be thought of as postponed indefinitely.)

The theoretical status of the deferred measurement principle had remained somewhat inconclusive. Available textbooks (e.g. [NC10, §4.4]) and numerous authors following them are content with inspecting a couple of examples while leaving it open what precisely the principle should state in generality, a situation recently criticized in [GB22a, §1]. A more precise form of the deferred measurement principle is briefly indicated in [Sta15, p. 6] and proposed there as an "axiom" of quantum computation. We prove below (Prop. 4.16) that the deferred measurement principle (9) is verified in the data-typing of quantum processes provided in `LHoTT`.

Notice that the content of this *equivalence between intermediate and deferred measurement collapse* (9) is not trivial without a good formalization; in fact it has historically been perceived as a *paradox*, namely this is essentially the paradox of "*Schrödinger's cat*" (where the cat plays the role of the intermediate controlled quantum gate). Moreover, the same paradox, in different words, was influentially offered in [Ev57a, pp. 4] as the main argument against the "Copenhagen interpretation" and for the "many-worlds interpretation" of quantum physics (cf. Lit. 2.2). Note that our same formalism which proves (9) also proves the equivalence (7) of these two "interpretations".

qRAM Models. Classical computing in its familiar *universal* form is based, in one way or another, on the model of a *Random Access Memory* ("RAM", cf. (36) below), abstracted as a *Mealy machine* [Me55]:

$$\begin{array}{c}\text{read-in RAM}\\ \text{\& input data}\end{array} \;\; \mathrm{RAM}\times D \xrightarrow{\begin{array}{c}\text{program interacting with}\\ \text{Random Access Memory}\end{array}} \mathrm{RAM}\times D' \;\; \begin{array}{c}\text{write RAM}\\ \text{\& output data}\end{array} \tag{10}$$

Starting with [Kn96], authors envisioned that quantum computing should similarly support a "qRAM model" [GLM08a][GLM08b], the basic idea being that data in qRAM may form quantum superpositions and may coherently be read/written in this form. As with the deferred measurement principle above, existing literature discusses this concept not in general abstraction but by way of concrete examples (see for instance [Ar$^+$15, Fig. 9][PPR19, Fig. 1][PCG23, Fig. 4][4]). From these one gathers that a quantum circuit of *nominal* type $\mathcal{H} \to \mathcal{K}$ but with access to a qRAM Hilbert space QRAM is *de facto* a quantum circuit of this form (a "circuit-based qRAM" [PPR19]):

$$\begin{array}{c}\text{read-in qRAM}\\ \text{entangled with}\\ \text{input quantum data}\end{array} \;\; \mathrm{QRAM}\otimes\mathcal{H} \xrightarrow{\begin{array}{c}\text{quantum program}\\ \text{interacting with qRAM}\end{array}} \mathrm{QRAM}\otimes\mathcal{H}' \;\; \begin{array}{c}\text{write qRAM}\\ \text{entangled with}\\ \text{output quantum data}\end{array} \tag{11}$$

In §4.3 we obtain (149) a formalized account/typing of qRAM and its equivalence to controlled quantum circuits.

Literature 2.2 (Epistemology of quantum physics and its formalization). The curious epistemology[5] of quantum physics occupied already the founding fathers of quantum theory [EPR35][Bohr1949] and the philosophical attitudes towards them were eventually canonized as *interpretations of quantum physics* [Me73][Sche73]. Later experimental advances in quantum physics only verified the nature of the theory and thus reinforced the epistemological puzzlement [GRZ99].

Quantum measurement. Concretely, the core issue is that what otherwise appears to be the epistemologically complete *state* of a quantum system – traditionally denoted "$|\psi\rangle$", being an element of some Hilbert space $\mathcal{H}$ – determines in general only the *probability* of which measurement outcome $w : W$ (which "world") will be observed upon measuring a given property of the system, while only *right after* the observation of a given w the quantum state appears to have "collapsed" along its linear projection onto a subspace of states with definite property w ([vonNeumann1932, §III.3, §VI][Lü51], cf. [Sche73, §IV][Om94, pp. 82][Re22, (A.2)]):

$$\begin{tikzcd}[row sep=small]
 & & \mathcal{H}_{w_1} & \begin{array}{c}\text{space of quantum states}\\ \text{with definite property } w_1\end{array} \\
\begin{array}{c}\text{Hilbert space of all}\\ \text{quantum states}\\ \text{of the given system}\end{array} & \mathcal{H} \;\simeq\; \underset{\begin{array}{c}\text{direct sum decomposition}\\ \text{in measurement basis } W\end{array}}{\Box_W \mathcal{H}_\bullet \;\equiv\; \bigoplus_{w':W} \mathcal{H}_{w'}} \arrow[ur, "\text{linear projection}"] \arrow[dr, "\text{linear projection}"'] & \vdots & \\
 & & \mathcal{H}_{w_n} & \begin{array}{c}\text{space of quantum states}\\ \text{with definite property } w_n\end{array}
\end{tikzcd} \tag{12}$$

To some extent this "state collapse" is formally just as expected (cf. [Ku05, §1.2][Yu12]) in a classical but probabilistic theory, where measurement of a random variable leads one to adjust the subjectively expected probability distribution according to Bayes' Law for updating conditional probabilities — except that *Kochen-Specker-Bell theorems* (e.g. [CS78][Ku05, §1.6.2][Mo19, §5.1.2]) show that (under very mild assumptions) generally no actual classical probability distribution can underlie a pure quantum state, hence that quantum states are *not* just a stochastic approximation to a more fundamental classical reality (cf. [Sche73, p. 140]).

Moreover, it seems untenable to regard the "state collapse" as just a subjective adjustment of expectation, since it is an operational component of experimentally realizable quantum communication protocols (cf. Lit. 2.1 and §4.3, such as in the *quantum teleportation* protocol recalled in §6.2); so much so that there is a paradigm of *measurement-only* quantum computation (cf. [Nie03][BBDRV09][Wei21]) where the computational process consists entirely of a sequence of such measurement-induced state collapses — in this practical sense the state collapse (12) *is an objective reality.*

Quantum epistemologies. The debates on what to make of the situation continue to this day (from the vast literature, see for instance [Om94][Bo08]), whence practicing physicists tend to just disregard the epistemological issue, an attitude that became proverbial under the catch-phrase "shut up and calculate" [Me89].
Among the main attitudes of quantum philosophers towards the issues are:

---

[4] A transparent example is discussed at `https://quantumcomputinguk.org/tutorials/implementing-qram-in-qiskit-with-code`

[5] Here "epistemology" – the *theory of kowledge* – refers to what can *in principle* (cf. [Fi07, p. 121]) be known about the (quantum) universe or any model or part of it, say about a given (quantum) computing machine, which in practice concerns the question of what can *in principle* be computed with a given quantum protocol, all imperfections of experiments and of experimenters disregarded.

- **Copenhagen epistemology: Quantum/classical divide.** The original "Copenhagen interpretation" (e.g. [Pr83, p. 99][Om94, p. 85]) pronounces a conceptual *frontier* or *divide* between quantum objects and their classical observers according to which recognizable result of any quantum measurement are, and must be reasoned about as, classical states.
- **Everett's epistemology: Branching into Many worlds.** An increasingly popular "many-worlds interpretation" (following H. Everett [Ev57a][Ev57b][dWG73]) rejects a separate classical component of quantum theory and instead asserts (informally and hence ambiguously, cf. [Te98]) both that the quantum state does never "really" collapse and at the same time that the universe successively "branches" into "many-worlds" inside which it nonetheless "appears" to observers to have collapsed in all possible ways.

The reader uneasy with making sense of any of this we invite to §4, where we present a *modal quantum logic* (cf. Lit. 2.13) which arguably makes precise these two epistemological attitudes and as such allows to prove their equivalence, cf. (7). In particular, the perceived paradox which Everett offers [Ev57a, pp. 4] to dismiss the Copenhagen interpretation and to motivate the "many-worlds" interpretation is arguably resolved by the *deferred measurement principle* (9), which becomes *provable* in quantum modal logic (Prop. 4.16).

Many possible worlds. Previously, several authors (e.g. [Bu76][Sk76, §III][Ta00, p. 101][No02, p. 22][Gi03, §8][Ter19][Wi20][AA22]) have vaguely wondered about or suggested a relation between these "many worlds" of quantum epistemology and the "possible worlds" in the sense classical modal logic (Lit. 2.13) but no formalized such discussion has previously been proposed. In particular, no previous author has considered this question with respect to a *linear* modal logic (cf. Lit. 2.4). (Beware that philosophers also speak of a *modal interpretation of quantum mechanics*[6] which shares some similarity in vocabulary but does not refer either to modal logic nor to many-worlds.)

The need for formalization. Indeed, in the time-honored spirit of Galileo, Kant, Hilbert, Wigner ("The book of nature is written in the language of mathematics.") one may have suspected that the fault causing epistemological troubles is not with quantum theory itself, but with speaking about it in ordinary informal language (Bohr 1920: "When it comes to atoms, language can only be used as in poetry."), whence their resolution lies instead in adopting a *mathematical* language of *non-classical formal logic* more appropriate for expressing microscopic quantum reality. In fact, a universal quantum programming language should essentially be just such a formal language, and in formulating it we do need to find a way to formally reflect the phenomenon of quantum measurement:

The verified programming of a quantum algorithm
is the act of accurately recounting in formalized language
the physical quantum process that executes it, and conversely.

It is towards this practical goal that here we care about quantum epistemology and may explain why we have more to say about quantum physics beyond quantum computation

Bohr toposes. Another proposal in the direction of formalized quantum epistemology may be recognized in [AC95] (in parallel and independently to the development of quantum/linear logic, Lit. 2.4). A variant of this proposal that gained some popularity is to use the internal logic of canonically ringed (co)presheaf toposes over the site of commutative subalgebras of a given C^*-algebra of quantum observables ("Bohr toposes", following ideas of [BHI98], for review see [Nui12][La17, §12]). The achievement of this approach is to show that the step from classical/commutative to quantum/noncommutative probability theory (of which a good account is in [Gl09][Gl11]) may be understood as the logical *internalization* of the classical axioms into a Bohr topos [HLS02]. While conceptually quite satisfactory, the practical relevance of this perspective has arguably remained elusive. In particular, it does not readily translate to a formal quantum (programming) language.

The approach which we take below is also ultimately (higher) topos-theoretic but otherwise rather complementary to Bohr toposes. In fact, one may understand Bohr toposes as formalizing the *Heisenberg picture* of quantum physics – where conceptual primacy is given to the algebras of *quantum observables* – while here we are concerned with the equivalent but "dual" *Schrödinger picture* where the primary concept is the spaces of *quantum states*: These being exactly the *linear types* that give this article its title. We indicate the connection to algebras of observables below in §5 but a detailed discussion needs to be given elsewhere.

Literature 2.3 (Topological quantum computation).
(For extensive motivation, explanation and referencing of topological quantum computation see the companion article [TQP].) The practical promise of quantum computation (Lit. 2.1) hinges on the achievability of fairly

---

[6] Cf. `plato.stanford.edu/entries/qm-modal`

undisturbed quantum processors which are sufficiently *robust* against the inevitable interaction with their environment. There are essentially two approaches toward robust quantum computation:

- **(i) Quantum error correction**: Operate on error-prone quantum hardware, but with software that implements enough redundancy to allow reading intended signals out of noisy background (cf. §6.3).
- **(ii) Topological error protection**: Operate on intrinsically stable quantum hardware which prevents errors from occurring in the first place.

In all likelihood, the eventual practice will be a combination of both approaches, since topological hardware error-protection achievable in the laboratory will itself have imperfections. Conversely, some quantum-error correction algorithms essentially consist of *simulating* topological quantum hardware on non-topological hardware, e.g. [Iq$^+$23]. However, the peculiarities of topological quantum gates had previously no genuine representation in quantum programming languages and were principally un-verifiable (cf. Lit. 2.4) until we argued, in the companion article [TQP], that realistic topological quantum gates are naturally modeled by *homotopy typed languages* (Lit. 2.6), such as classical `HoTT` and, more accurately, by `LHoTT` (§3).

Literature 2.4 (**Formal (quantum) software verification and dependent (linear) data typing).**
(For extensive exposition and referencing of the *classical* case see the companion article [TQP].)
The benefit or even necessity of *formal software verification methods* [CC09][Me11] (often abbreviated to just "formal methods", cf. [WLBF09]) — hence of computer-checked proof at compile-time of correct behavior of critical software — is evident [HN19] and as such increasingly of interest for instance to the crypto-reliant industry (e.g. [Hed18][VYC22][Qu23]) and the military (e.g. MURI:FA95501510053). Nevertheless, in less critical applications of classical computation the overhead associated with formal verification is still widely traded for the possibility of incrementally de-bugging faulty software during application.

Need for verification of quantum programs. However, such run-time debugging is no longer a sustainable option when it comes to serious *quantum* computation, due ([VRSAS15, p. 6][FHTZ15][Ra18][7][YF18][MZD20][YF21]) to its:

- drastically higher complexity,
- drastically higher run-time cost,
- impossibility of run-time inspection.

The last point is the fundamental one, enforced by the quantum laws of nature (state collapse under measurement, Lit. 2.2), but the other two points will in practice be no less forbidding.

Accepting the need for (quantum) software verification, its implementation of choice is by *data typing* (which for quantum data means "dependent linear typing" discussed in §3):

Formal verification by data typing. A profound confluence of computer science and pure mathematics occurs with the observation [ML82] that formal software verification is not only amenable to constructive mathematical proof but fundamentally equivalent to it – every constructive mathematical proof may be understood as pseudocode for a program whose output is data of the type of certificates of the truth of the given statement, a profound tautology known as the *BHK (Brouwer–Heyting–Kolmogorov) correspondence*, or similar.

Accordingly, formal verification/proof languages are (dependently) *typed* in that every piece of data they handle has assigned a precise *data type* which provides the strict specification that data has to meet in order to qualify as input or output of that type ([ML82][Th91][St93][Lu94][Gu95][Co11][Ha16]). The abstract theory of such data typing is known as (dependent-)*type theory* and the modern flavor relevant here is often called *Martin-Löf type theory* in honor of [ML71][ML75][ML84]; for more elaboration and introduction see also [Ho97][UFP13].

Once this typing principle is adhered to, the distinction vanishes between writing a program and verifying its correctness. Moreover, such a properly typed functional program may equivalently be understood as a *mathematical* object, namely as a mathematical function (13) from the "space" of data of its input type to that of its output

---

[7] [Ra18, p. iv]: "We argue that quantum programs demand machine-checkable proofs of correctness. We justify this on the basis of the complexity of programs manipulating quantum states, the expense of running quantum programs, and the inapplicability of traditional debugging techniques to programs whose states cannot be examined. [...] Quantum programs are tremendously difficult to understand and implement, almost guaranteeing that they will have bugs. And traditional approaches to debugging will not help us: We cannot set breakpoints and look at our qubits without collapsing the quantum state. Even techniques like unit tests and random testing will be impossible to run on classical machines and too expensive to run on quantum computers – and failed tests are unlikely to be informative. [...] Thesis Statement: *Quantum programming is not only amenable to formal verification: it demands it.*"

type — called its *denotational semantics* (a seminal idea due to [Sc70][ScSt71]; for exposition see [SK95, §9]):

Syntax	**Semantics**
program $\overset{\text{flags}}{\gamma{:}\Gamma},\ \overset{\text{input}}{i{:}I} \ \vdash\ \overset{\text{output}}{p_\gamma(i) : O}$	function $\overset{\text{domain}}{\Gamma \times I} \xrightarrow{\ \vdash p\ } \overset{\text{codomain}}{O}$

(13)

For classical[8] data types the *inference rules* by which such program/function declaration may proceed equip the type universe with the structure of a Cartesian closed category [LS86, §I], whence one also speaks of *categorical semantics* (see [Ja98][Ja93]). Here the inference rules for the classical logical conjunction "×", hence for the Cartesian product, subsume the basic "structural rules" called the *contraction rule* and the *weakening rule* (e.g. [Ja94][Ja98, p. 122][UFP13, §A.2.2][Rij18, §1.4]), which semantically express the possibility of duplicating and of discarding classical data:

structural inference rules for classical data types	**Syntax**	**Semantics**
	C $\dfrac{\Gamma,\ p_1{:}P,\ p_2{:}P \ \vdash\ t_{p_1,p_2} : T}{\Gamma,\ p{:}P \ \vdash\ t_{p,p} : T}$ Contraction rule	$\dfrac{\Gamma \times P \times P \xrightarrow{\ \vdash t\ } T}{\Gamma \times P \xrightarrow{\mathrm{id}_\Gamma \times \mathrm{diag}_P} \Gamma \times P \times P \xrightarrow{\ \vdash t\ } T}$ Diagonal (cloning)
	W $\dfrac{\Gamma \ \vdash\ P : \mathrm{Type} \qquad \Gamma \ \vdash\ t : T}{\Gamma,\ P \ \vdash\ t : T}$ Weakening rule	$\dfrac{\Gamma \xrightarrow{\ \vdash t\ } T}{\Gamma \times P \xrightarrow{\ \mathrm{pr}_\Gamma\ } \Gamma \xrightarrow{\ \vdash t\ } T}$ Projection (deletion)

(14)

The quest for quantum data typing was historically convoluted (starting with the much debated quantum logic of [BvN36]) but is, in hindsight, fairly straightforward: Since the hallmark of coherent quantum evolution is (see [Ab09] for a structural account) the pair of:

- the *no-cloning theorem* ([WZ], saying that quantum data cannot be *systematically* duplicated),
- the *no-deletion theorem* ([PB00], saying that quantum data cannot be *systematically* discarded),

it follows that a program handling purely quantum data types must *not* use the structural rules (14) for the logical conjunction of quantum data, which is then called the (non-Cartesian) *tensor product* ⊗. It is this *removal* of structural inference rules ("sub-structural logic") which frees the tensor product of quantum data types from only consisting of pairs of data and hence allows for the hallmark phenomenon of *quantum entanglement*.

Such *sub-structural* languages were essentially introduced in (the "multiplicative fragment" of) the *linear logic* (see [Se89][MN13]) of [Gir87] (who was apparently vaguely aware of potential application to quantum logic, cf. [Gir87, p. 7]). These languages were then suggested as expressing quantum processes in [Ye90][Pr92] and were more fully understood as quantum (programming) languages (Lit. 2.5) with *linear types* in [Val04][SV05] [AD06][Du06][SV09]. Notice that the adjective "linear" here refers to the preservation of the number of type factors in the absence of the structural rules (14), which implies that functions $f : X \to Y$ between linear types must indeed use their argument $x : X$ linearly, in the algebraic sense.

Quantum Phenomena	**Linear Type Inference**
No-cloning theorem	Absence of contraction rule
No-deleting theorem	Absence of weakening rule

(15)

This resulting principle that

Quantum data has linear type.

has meanwhile come to be more commonly appreciated (e.g. [DLF12, pp. 1]) in particular in quantum language design (Lit. 2.5, [FKS20]):

---

[8] Here by *classical types* we mean the types of *intuitionistic* Martin-Löf type theory in contrast to *linear* (quantum) types (15), but *not* in the sense of "classical logic": Classical types in our sense are *not quantum* in that they are subject to the structural inference rules (14) but they are still *constructive* in that they are not (necessarily) subjected to the law of excluded middle and/or the axiom of choice (which distinguish classical logic from constructive logic).

[Sta15]: *A quantum programming language captures the ideas of quantum computation in a linear type theory.*

Bunched classical/quantum type theory and EPR phenomena. And yet, a comprehensive programming language implementing such *linear type theories* of *combined* classical and quantum data had remained elusive all along: The type-theoretic subtlety here is that with the classical conjunction $(\times)$ being accompanied by a linear multiplicative conjunction $(\otimes)$, then contexts on which terms and their types should depend are no longer just linear lists of (dependent) classical products

$$\Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_n \qquad \begin{array}{c}\textbf{a classical type-context} \\ \textbf{(tuples of classical data)}\end{array}$$

but may be nested ("bunched") such products, alternating with linear multiplicative conjunctions to form tree-structured expressions like this example:

$$\Gamma_1 \times \big(\Gamma_2 \otimes (\Gamma_3 \times \Gamma_4)\big) \times (\Gamma_5 \otimes \Gamma_6) \times (\Gamma_7 \otimes \Gamma_8 \otimes \Gamma_9) \qquad \begin{array}{c}\textbf{a mixed classical/quantum type-context} \\ \textbf{(tuples of classical data mixed with } \textit{entangled} \textbf{ quantum data)}\end{array} .$$

While the idea of formulating such "bunched" type theories is not new [OP99][Py02][O'H03], its implementation has turned out to be tricky and the results unsatisfactory; see [Py08, §13.6][Ri22, p. 19]. The claim of the type theory introduced in [Ri22] is to have finally resolved this long-standing issue of formulating "bunched linear dependent type theory". Here we understand this as saying that a verifiable universal quantum programming language now exists (`LHoTT`, §3).

To put this into perspective it may be noteworthy that the root of this subtlety resolved by `LHoTT` corresponds to the hallmark phenomenon of quantum physics which famously puzzled the subject's founding fathers (Lit. 2.2), namely the *conditioning of physics on entangled quantum states* (known as the *EPR phenomenon*, e.g. [Sel88]):

Under the correspondence between dependent linear type theory and quantum information theory, the existence of bunched typing contexts involving linearly multiplicative conjunctions ⊗ corresponds to the conditioning of protocols on entangled quantum states and hence to what in quantum physics are known as EPR phenomena.

Bunched logic	**EPR phenomena**
Typing contexts built via multiplicative conjunction (⊗)	Physics conditioned on entangled quantum states

Exponential modality. In previous lack of a classically-dependent linear type theory, the strategy for recovering classical logic among a linear (quantum) type system was to postulate a modal operator (Lit. 2.13) on the linear type system – traditionally denoted "!" [Gir87] and (sometimes) called the *exponential modality* – where a linear type of the form $!\mathcal{H}$ may be thought of (cf. Rem. 3.9 below) as behaving like the linear span of the *underlying set* of a linear space $\mathcal{H}$, thus giving the linear type system a kind of access to this underlying classical type. Eventually it came to be appreciated (cf. [Me09, p. 36]) that the exponential modality should (this is due to [Se89, §2] and [dP89][BBdP92, §8][BBdPH92]) be axiomatized as a comonad (cf. Lit. 2.15) and specifically as a comonad induced by a suitably monoidal adjunction between linear and classical (intuitionistic) types (due to [Bi94, pp. 157][Be95])

$$\begin{tikzcd}[column sep=large]
\text{ClaType} \arrow[r, shift left=2, "\overset{\textbf{quantization}}{Q}"] \arrow[r, phantom, "\perp"] & \text{QuType} \arrow[l, shift left=2, "\underset{\textbf{classicization}}{C}"] \arrow[loop right, "!\ \textbf{exponential modality}"] \\
\begin{array}{c}\textbf{purely} \\ \textbf{classical} \\ \textbf{(intuitionistic)} \\ \textbf{types}\end{array} & \begin{array}{c}\textbf{purely} \\ \textbf{quantum} \\ \textbf{(linear)} \\ \textbf{types}\end{array}
\end{tikzcd} \tag{16}$$

Traditionally, inference rules for such an exponential modality need to be adjoined to plain (non-dependent) linear type theories, which is laborious and not without subtleties ([Gir93][Wa93][Be95][Ba96]). In contrast, in Prop. 3.8 we obtain (cf. [Ri22, Prop. 2.1.31]) an exponential modality from the basic type inference provided by a *dependent* linear type theory like `LHoTT`, a possibility first highlighted in [PS12, Ex. 4.2][Sch14b, §4.2].

Full verification: Towards identity types. Either way, (linear) data-typing in general serves to impose and verify consistency constraints on (quantum) data. But for a fine-grained certification of program behaviour by *equational* constraints — eg. for certifying correctness of quantum teleportation protocols (cf. Rem. 6.2) or of quantum error corrections (cf. Rem. 6.3) — one specifically needs certificates of *identification types* (colloquially: "identity types"), certifying the (operational) equality of pairs of data of a given type.

But the correct formal treatment of data types of identifications turns out to be surprisingly subtle, which may be one reason why none of the previously existing quantum programming languages provide such identity types — and this includes the typed functional languages `QML` and Proto-`Quipper`, cf. Lit. 2.5. Namely, once identifications of any data pairs $d, d' : D$ are promoted to data of identification type $p : \mathrm{Id}_D(d,\, d')$ ("propositional equality"), the same principle applies to pairs $p, p' : \mathrm{Id}_D(d\,d')$ of these certificates themselves, whose verifiable identification now requires data of *iterated identification type* $\mathrm{Id}_{\mathrm{Id}_D(d,d')}(d,\, d')$ — *and so on.* The proper handling of this phenomenon requires and leads *homotopy types* of data provided by classical `HoTT` and its linear form `LHoTT`, see the discussion in Lit. 2.6.

Literature 2.5 (Quantum programming languages). The idea of quantum programming languages was first systematically expressed in [Kn96], early proposals for formalization (via a kind of linear types, Lit. 2.3) are due to [Se04][Val04][SV05][SV09]. Exposition of the need and relevance of quantum programming languages (which was not originally obvious to the community, cf. the historical lead-in to [Se16]) specifically for quantum/classical hybrid computation, may be found in [VRSAS15]. Based on these early developments (and besides a multitude of quantum circuit languages that now exist for programming the available NISQ machines, Lit. 2.8), currently there exists essentially one quantum programming language with universal ambition: `Quipper`[9] [GLRSV13][GLRSV13] (for exposition see [Se16]). In its formalized fragment called "Proto-`Quipper`" [Ro15, §8][RS18, §4.3] this language may be understood as involving a kind of dependent linear types, Lit. 2.4) with semantics in categories of indexed sets of linear objects ([RS18][FKS20][Lee22][Ri21]), notably in indexed sets of (complex) vector spaces, of the same kind as that in (3) we discuss as semantics for the 0-fragment (Rem. 2.11) of `LHoTT`.

Literature 2.6 (Homotopically typed languages). (For more extensive review of this point see the companion article [TQP].)
An operation on data so fundamental and commonplace that it is easily taken for granted is the *identification* of a pair of data with each other. But taking the idea of program verification by data typing (Lit. 2.4) seriously leads to consideration also of *certificates of identification* of pairs of data of any given type which thus must themselves be data of "identification type" [ML75, §1.7].

Trivial as this may superficially seem, something profound emerges with such "thoroughly typed" programming languages (the technical term is: *intensional type theories* (see [St93, p. 4, 13][Ho95, p. 16]) in that now given a pair of such identification certificates the same logic applies to these and leads to the consideration of identifications-of-identifications (first amplified in [HS98]), and so on to higher identifications, *ad infinitum.*

Remarkably, the "denotational semantics" (Lit. 2.4) of data types equipped with such towers of identification types, hence the corresponding pure mathematics, is ([AW09][Aw12], exposition in [Sh12][Ri22]) just that of abstract homotopy theory (Lit. 2.10) where identification types are interpreted (**??**) as path spaces and higher-order identifications correspond to higher-order homotopies. One also expresses this state of affairs, somewhat vaguely, by saying that HoTT has *semantics* in homotopy theory, and conversely that HoTT is a *syntax* for homotopy theory – we have reviewed this dictionary in [TQP, §5.1].

Ever since this has been understood, the traditional ("intuitionistic Martin-Löf"-)type theory of [ML75][NPS90] has essentially come to be known as *homotopy type theory* (HoTT) – specifically so if accompanied by one further "univalence" axiom[10] (for more on this see the companion article around [TQP, (105)]) which enforces that identification of data types themselves coincides with their operational equivalence (exposition in [Ac11]).

The standard textbook account for "informal" (human-readable) `HoTT`is [UFP13], exposition may be found in [BLL13], gentle introduction in [Rij18][Rij23] (the former more extensive); and see the companion article [TQP], Section 5. Available software that *runs* homotopically typed programs includes `Agda`[11] and `Coq`[12].

Literature 2.7 (Topological quantum compilation.). Once serious quantum computation hardware (Lit. 2.3) becomes available, a central effort in quantum computation (Lit. 2.1) concerns *quantum compilation* [MMRP21], namely in translating high-level quantum algorithms into sequences (circuits) of logic gate operations which the hardware actually implements. The seminal *Solovay-Kitaev theorem* ([NC10, App. 3][DN06]) guarantees, under rather mild assumptions on the available gate set, that such a compilation is always possible, but optimization for scarce runtime resources requires considerable effort.

The problem of quantum computation is particularly demanding for topological quantum computation (Lit. 2.3), hence in the case of *topological quantum compilation* (e.g. [HZBS07][Br14][KBS14]), since here the available gate logic is far remote from then QBit-based operations (8) in which high-level quantum algorithms are conceived. No attempt seems to previously have been made toward formally verifying a topological quantum compilation, and indeed the problem is not captured by classical verification strategies. Notice that:

---

[9] Landing page: `www.mathstat.dal.ca/∼selinger/quipper`

[10] The univalence axiom is widely attributed to [Vo10], but the idea (under a different name) is actually due to [HS98, §5.4], there however formulated with respect to a subtly incorrect type of equivalences (as later shown in [UFP13, Thm. 4.1.3]). The new contribution of [Vo10, p. 8, 10] was a good definition of the types of ("weak") equivalences between types.

[11] `Agda` landing page: wiki.portal.chalmers.se/agda/pmwiki.php

[12] `Coq` landing page: `coq.inria.fr`

(i) formal verification of quantum compilation, in general, is not a discrete but an *analytical* problem, whose computer verification requires *exact real (complex) computer arithmetic* (cf. [TQP, Lit, 2.29]),

(ii) the generic topological quantum gate is given by a complicated analytical expression (cf. [TQP, Lit. 2.24]).

While here we will not further dwell on the issue explicitly, the claim of [TQP] is that these two problems are addressed by homotopically-typed certification languages (`HoTT`, Lit. 2.6) of which the language `LHoTT` of concern here is an extension.

Literature 2.8 (NISQ computers). Currently existing quantum computers (such as those based on "superconducting qbits", see e.g. [CW08][HWFZ20]) serve as proof-of-principle of the idea of quantum computation (Lit. 2.1) but offer puny computational resources, as they are (very) **n**oisy and (at best) of **i**ntermediate **s**cale: "NISQ machines" [Pr18][LB20]. What is currently missing are noise-protection mechanisms that would allow to scale up the size and coherence time of quantum memory. The foremost such protection mechanism arguably (Lit. **??**) is *topological* protection (Lit. 2.3).

Literature 2.9 (Classically controlled quantum computation and dynamic lifting).

classical control [NPW07][De14]

the term "dynamic lifting" is due to [GLRSV13, p. 5], early discussion is in [Ra18, pp. 40]. Proposals for its categorical semantics are discussed in [RS20][LPVX21][FKRS22][FKRS22][CDL22][Lee22].

Literature 2.10 (Parameterized stable homotopy theory, Tangent ∞-toposes & Twisted cohomology). One may observe that the following two fundamental types of 1-categories (cf. 2.4):

(i) *toposes* – which are the home of geometry and classical intuitionistic logic,

(ii) *abelian categories* – which are the home of linear algebra and forms of linear logic,

while antithetical (for instance in that only the terminal category is an example of both), secretly share a sizeable list of exactness properties [Fr99]. The analogous situation for ∞-categories may appear similar, since here the two notions of

(i) ∞*-toposes* – which are the home of higher geometric and of classical (intuitionistic) homotopy type theory,

(ii) *stable* ∞*-categories* – which are the home of higher algebra,

do remain as antithetical, (even though both satisfy analogous Giraud-type axioms in that both arise, when locally presentable, as accessible left-exact localizations of ∞-categories of presheaves: the former with values in ∞-groupoids, the latter with values in spectra).

But a miracle happens after the passage to ∞-category theory, in that here a non-trivial unification of the two notions does exist for a large class of stable ∞-categories ("Joyal loci") including those of module spectra. Namely, the collection of *parameterized spectra* [MaSi06][Mal23] over varying base types $\mathcal{X} \in \mathrm{Grpd}_\infty$ — i.e., the ∞-Grothendieck construction on the ∞-functor categories to $R\mathrm{Mod}(\mathrm{Spctr})$ — is itself an ∞-topos:

$$R \,\in\, E_\infty\mathrm{Ring}\big(\mathrm{Spctr}\big) \qquad\qquad \vdash \qquad\qquad T^R\mathrm{Grpd}_\infty \;:\equiv\; \int_{\mathcal{X}\in\mathrm{Grpd}_\infty} R\mathrm{Mod}^{\mathcal{X}} \;\;\in\; \mathrm{Topos}_\infty\,. \tag{17}$$

This observation is originally due to [Bie07], was noted down in [Jo08, §35] and received a dedicated discussion in [Ho19]. The special case for plain spectra (i.e. with $R = \mathbb{S}$ the sphere spectrum), is touched upon in [Lu17, Rem. 6.1.1.11], where $\int_{\mathcal{X}} \mathrm{Spectra}^{\mathcal{X}}$ would be called the *tangent bundle* to Grpd_∞ [Lu17, §7.3.1] when thought of as equipped with the canonical projection to the base topos (19). We may thus think of (17) as something like the R*-linear tangent* ∞*-topos* to Grpd_∞ [Sch13, Prop. 4.1.8] (all these considerations work for base ∞-toposes other than Grpd_∞; which we disregard just for sake of exposition).

Twisted cohomology. Interestingly, the hom-spaces in the R-tangent ∞-topos (17) are sections of R-module bundles $\tau_{\mathcal{X}}$, which means [ABGHR14][FSS23, Prop. 3.5][SS20, p. 6] that their connected components form the $\tau_{\mathcal{X}}$*-twisted* R*-cohomology* $R^\tau(\mathcal{X})$ of $\mathcal{X}$ [MaSi06, §22.11]:

$$\left.\begin{array}{l} \mathcal{X} \,\in\, \mathrm{Grpd}_\infty \\ R \,\in\, E_\infty\mathrm{Rng}(\mathrm{Spctr}) \end{array}\right\} \qquad \vdash \qquad \mathrm{Maps}\big(0_{\mathcal{X}},\, R /\!\!/ \mathrm{GL}_1(R)\big) \;\;=\;\; \left\{ \begin{tikzcd} & R /\!\!/ \mathrm{GL}_1(R) \arrow[d, two heads] \\ \mathscr{X} \arrow[ur, dashed, "\text{cocycle in } R^{\tau}(\mathcal{X})"] \arrow[r, dashed, "\tau_{\mathcal{X}}" description, "\text{twist}"'] & B\mathrm{GL}_1(R) \end{tikzcd} \right\}. \tag{18}$$

This already suggests [Sch14a] that tangent ∞-toposes are a natural logical context for describing strongly-coupled quantum systems, since twisted R-cohomology theories play a key role in their holographic (stringy) formulations (Lit. ...).

To pinpoint the nature of this logical context, notice that there is a canonical inclusion of Grpd_∞ into its tangent ∞-topos (17) by assigning the 0-spectrum everywhere. Since the 0-spectrum is a zero-object, it readily follows that this inclusion is bireflective in that it is both left and right adjoint to the "tangent projection"

$$\begin{tikzcd}
\textbf{R-linear tangent ∞-topos} & T^R\mathrm{Grpd}_\infty \arrow[loop above, "\overset{\textbf{classical modality}}{\natural}"] \arrow[d, shift right=4, "p"'] \arrow[d, shift left=4, "p"] \arrow[d, phantom, shift right=2, "\dashv"] \arrow[d, phantom, shift left=2, "\dashv"] & = \int_{\mathcal{X}} R\mathrm{Mod}^{\mathcal{X}} & \textbf{flat ∞-bundles of R-module spectra} \\
\textbf{classical base ∞-topos} & \mathrm{Grpd}_\infty \arrow[u, hook, "0"] & &
\end{tikzcd} \tag{19}$$

In [Sch13, Prop. 4.1.9] this situation is interpreted as exhibiting *infinitesimal cohesive structure* on $T^R\mathrm{Grpd}_\infty$ relative to Grpd_∞, meaning that, in some precise abstract sense, the objects of $\mathrm{T}^R\mathrm{Grpd}_\infty$ may be regarded as equipped with an *infinitesimal thickening* of sorts: In the notation there, the adjoint pair of (co)monads induced by the adjoint triple (19) is denoted $\int \dashv \flat$, expressing the *shape* and the *underlying points* of an object, respectively; and the ambidexterity of the adjunction implies that the canonical *points-to-pieces transform* is an equivalence

$$\flat \xrightarrow{\sim} \int$$

hence reflecting the idea that the extra geometric substance which the objects of $\mathrm{T}^R\mathrm{Grpd}_\infty$ carry on their classical underlying skeleta in Grpd_∞ is "infinitesimal" (think: "microscopic") so that it cannot be noticed from looking just at the macroscopic shape of these objects.

As a result, these two modalities unify into a single ambidextrous modality which was denoted "$\natural$" in [RFL21], as shown in (19).

Remark 2.11 (0-Fragment of `LHoTT`). By the 0-fragment of `LHoTT` we mean more than just its 0-truncated types (which are just the classical hSets of `LHoTT`). Namely, in the *stable* homotopy theory which is incorporated in `LHoTT`, the classical notion of n-truncation becomes almost meaningless (due to the existence of spectra with homotopy groups in arbitrary *negative* degree, cf [Lu17, Warning 1.2.1.9]), its proper replacement instead being the notion of *t-structure* (eg. [Lu17, §1.2.1]). The *heart* of the t-structure (formed by the spectra whose homotopy groups are conceptrated in degree 0) reflects the intended 0-fragment of the given stable homotopy theory. Hence by the 0-fragment of `LHoTT` we mean those types which are in the heart and whose *underlying* purely classical type is 0-truncated.

Literature 2.12 (Functional languages). With all data being of specified type, a *program* which, when run on input data of type D_{in} (is guaranteed to halt and then) produces data of type D_{out} is thus a *function* of the collection of D_{in}-data with values in the collection of D_{out}-data, and we may postpone detailing what particular kind of function we might mean (for instance: *linear* functions for quantum programs) by speaking of just an arrow (morphism) in the relevant *category of types*:

Programming syntax			**Categorical semantics**		
$d : D_{\mathrm{in}}$	$\vdash$	$f(d) : D_{\mathrm{out}}$	D_{in}	$\xrightarrow{\;f\;}$	D_{out}
input data type	**program**	**output data type**	**domain object**	**morphism**	**codomain object**

The point of "functional" programming is that programs are such functions of data (morphisms) and *nothing but* such functions, in that they have no side-effects (besides producing their output) and no side-dependence (besides on their input) on the state of the computing environment — therefore also called *pure functions* or *pure programs*, for emphasis. This is in contrast to popular "imperative" programming languages — whose programs may, while running, read unpredictable data from input devices and write to output devices in a way that is not reflected in the specification of their input/output data types. Instead, the purity of functional programs is what makes them predictable and hence verifiable.

Literature 2.13 (**Modal logic and Possible worlds semantics**). The origin of modal logic of *necessity* ($\Box$) and *possibility* ($\Diamond$) is with Aristotle, as nicely reviewed in [LeS77]. The modern formalization of modal logics originates with [Be30][LL32, pp 153 & App II][vW51][Hi62]. A good historical overview is in [Go03], a comprehensive modern account in [BvBW07]; see also [BdRV01]. Starting with [LL32, App II], modal logicians consider a plethora of variant axiom systems, which go by a long list of alphanumerical monikers. We are here entirely concerned with the system known as "S5" modal logic [LL32, p. 501][Kr63, p. 1]. Classical S5 modal logic is widely applied as epistemic modal logic, notably in classical computer science [HM92, §2.3][FHMV95, p. 35][Fi07, §9][HP07, §4][DHK08, §2][Sa10].

Possible worlds semantics. The "possible worlds"-semantics of modal logic is due to [Kr63] (though the basic idea is expressed already in [Hi62]); good exposition is in [BvB07], modern review is in [BvBW07, Part 5 §1]. Here one speaks of *Kripke frames* being (inhabited) W : Set of "possible worlds" equipped with a binary relation $R : W \times W \to \mathrm{Prop}$, where $R(w, w')$ is interpreted as "Given outcome/world w, the outcome/world w' appears (just as) *possible*." Given such a possible-worlds scenario, the modal operators $\Box_W$, $\Diamond_W$: $\mathrm{Prop}_W \to \mathrm{Prop}_W$ acting on W-dependent propositions P : $\mathrm{Prop}_W \equiv W \to \mathrm{Prop}$ are interpreted by the following formulas (e.g. [BvB07, p. 10]):

$$\begin{array}{ccccc}
\begin{array}{c}\textbf{A proposition } P_\bullet\\ \textbf{about/dependent on}\\ \textbf{the possible worlds } w\end{array} & \text{yields} & \begin{array}{c}\textbf{The proposition } \Box_w P\\ \textbf{that } P_\bullet \textbf{ holds } \textit{necessarily}\textbf{, namely}\\ \textbf{in/for } \textit{all} \textbf{ worlds } w' \textbf{ that appear}\\ \textbf{as possible as the given one } w\end{array} & \text{and} & \begin{array}{c}\textbf{The proposition } \Diamond_w P\\ \textbf{that } P \textbf{ holds } \textit{possibly}\textbf{, namely}\\ \textbf{in/for } \textit{some} \textbf{ world } w' \textbf{ that appears}\\ \textbf{as possible as the given one } w\end{array} \\
\begin{array}{rccc} P_\bullet : & W & \longrightarrow & \mathrm{Prop} \\ & w & \mapsto & P_w \end{array} & \vdash & \begin{array}{rccc} \Box_W P : & W & \longrightarrow & \mathrm{Prop}, \\ & w & \mapsto & \underset{\substack{(w':W)\times \\ R(w,w')}}{\forall} P_{w'} \end{array} & & \begin{array}{rccc} \Diamond_W P : & W & \longrightarrow & \mathrm{Prop} \\ & w & \mapsto & \underset{\substack{(w':W)\times \\ R(w,w')}}{\exists} P_{w'} \end{array}
\end{array} \tag{20}$$

Modalities as monads. The (co)monadic nature of the necessity/possibility operators $\Box/\Diamond$ in S4 (hence in S5) modal logic was explicitly observed in [BdP96][BdP00][Ko97] and the resulting relation of modalities to (computational effect-)monads in computer science (Lit. 2.15) was further discussed in [BBdP98]. The natural origin of these S5 (co)monads $\Box_W \dashv \Diamond_W$ from *base change* along the "possible worlds" was noticed in [Aw06, p. 279] – however the implication (which we expand on in §4) that, therefore, any dependent type theory may equivalently be regarded as (epistemic) *modal type theory* (Lit. 2.14) seems not to have received attention until the note [*n*Lab14] (cf. [Co20, Ch. 4]). We expand on this novel point of view in the main text around Thm. 4.3.

Literature 2.14 (**Modal type theory**). In view of the famous relation between formal logic and type theory, it is quite evident that there is an interesting generalization of modal logic (Lit. 2.13) to *modal type theory*. After leading a niche existence for some time, the amplification [Sch13, §3.1][ScSh14] of *cohesive* modalities (see [SS20]) in (homotopy) type theory, the subject of *modal type theory* has received much attention (e.g. [RSS20][CR21][Mye22]). While such modal type theory is going to be relevant for various enhancements of the computational context presented here (to be discussed elsewhere), we emphasize that the modalities we consider here are all provided already by plain (linear) dependent type theory. This fact is what drives our observation that `LHoTT` already knows about quantum measurement effects – the feature just has to be brought out by meticulous syntactic sugaring.

Literature 2.15 (**Computational Effects and Logical Modalities**). We give a lightning explanation of computational effects (and computational contexts) understood as (co)monads on the type system, and of the Eilenberg-Moore-Kleisli theory of the corresponding effect handlers (context providers) understood as (co)modules, in fact as (co)modal types (cf. Lit. 2.14).

Computational effects as Monads on the type system. The idea ([Mog89][Mog91][PP02], cf. [HP07, §6]) is that a computation which *nominally* produces data of some type D while however causing some computational side-effect must *de facto* produce data of some adjusted type $\mathcal{E}(D)$ which is such that the effect-part of the adjusted data can be carried alongside followup programs (whence a "notion of computation" with "computational side effects", for exposition and review see [BHM02][Mi19, §20][Uu21]):

$$\begin{tikzcd}[column sep=large]
D_1 \arrow[r, "\overset{\textbf{first program}}{\mathrm{prog}_{12}}", "\substack{\textbf{output data}\\ \textbf{of nominal type } D_2\\ \textbf{causing effects of type } \mathcal{E}(-)}"'] & \mathcal{E}(D_2) & D_2 \arrow[r, "\overset{\textbf{second program}}{\mathrm{prog}_{23}}", "\substack{\textbf{input data}\\ \textbf{of type } D_2\\ \textbf{causing effects of type } \mathcal{E}(-)}"'] & \mathcal{E}(D_3) & {} \arrow[d, mapsto, bend left, "\substack{\textbf{bind previous effects}\\ \textbf{into second program}}"] & D \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D}", "\textbf{return plain data with trivial } \mathcal{E}(-)\textbf{-effect}"'] & \mathcal{E}(D) \\
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathcal{E}(D_2) & \mathcal{E}(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}", "\substack{\textbf{carry any previous}\\ \mathcal{E}(-)\textbf{-effects along}}"'] & \mathcal{E}(D_3) & {} \arrow[d, mapsto, bend left, "\textbf{compose}"] & \mathcal{E}(D) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\big(\mathtt{return}^{\mathcal{E}}_{D}\big)", "= \mathrm{id}_{\mathcal{E}(D)}"'] & \mathcal{E}(D) \\
D_1 \arrow[rrr, "\overset{\mathcal{E}\textbf{-composite program}}{\big(\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}\big) \circ \mathrm{prog}_{12}}", "\textbf{causing cumulative } \mathcal{E}(-)\textbf{-effects}"'] & & & \mathcal{E}(D_3) & {} & &
\end{tikzcd} \tag{21}$$

Such $\mathcal{E}$-effect structure on the type system is equivalently [Ma76, p. 32][Mog91, Prop. 1.6] a functorial operation on the category of types (given by forming "effectless programs")

$$\begin{tikzcd}
\mathcal{E} \;: & \mathrm{Type} \arrow[rr, "\textbf{functor underlying monad}"] & & \mathrm{Type} \\
& \big(D_1 \xrightarrow{f} D_2\big) \arrow[rr, mapsto, "\textbf{regard } f \textbf{ as effectless program}"'] & & \mathtt{bind}^{\mathcal{E}}\Big(D_1 \xrightarrow{f} D_2 \xrightarrow{\mathtt{return}^{\mathcal{E}}_{D}} \mathcal{E}(D_2)\Big)
\end{tikzcd} \tag{22}$$

which carries the structure of a **monad**[13] (cf. [ML71, §VI][Bor94b, §4], older terminology: "triple"), namely natural transformations

$$D \xrightarrow{\overset{\textbf{monad unit}}{\mathrm{ret}^{\mathcal{E}}_{D} \equiv \mathtt{return}^{\mathcal{E}}_{D}}} \mathcal{E}(D) \qquad\qquad \mathcal{E}\big(\mathcal{E}(D)\big) \xrightarrow{\overset{\textbf{monad multiplication}}{\mathrm{join}^{\mathcal{E}}_{D} \equiv \mathtt{bind}^{\mathcal{E}}_{\mathrm{id}_{\mathcal{E}(D)}}}} \mathcal{E}(D) \tag{23}$$

satisfying the axioms of a unital monoid, in that they make the following natural diagrams commute

$$\begin{tikzcd}
\mathcal{E}(D) \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[d, "\mathcal{E}\left(\mathrm{ret}^{\mathcal{E}}_{D}\right)"'] \arrow[dr, equal, "\textbf{unitality}" description] & \mathcal{E}\big(\mathcal{E}(D)\big) \arrow[d, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"] & & \mathcal{E}\left(\mathcal{E}\left(\mathcal{E}(D)\right)\right) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[d, "\mathcal{E}\left(\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}\right)"'] \arrow[dr, phantom, "\textbf{associativity}"] & \mathcal{E}\big(\mathcal{E}(D)\big) \arrow[d, "\mathrm{join}^{\mathcal{E}}_{D}"] \\
\mathcal{E}\big(\mathcal{E}(D)\big) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"'] & \mathcal{E}(D) & & \mathcal{E}\big(\mathcal{E}(D)\big) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{D}"'] & \mathcal{E}(D)\,.
\end{tikzcd} \tag{24}$$

Monads induced by adjunctions. Monads arise from (cf. [ML71, §VI.1][Bor94b] – and also give rise to, see (45) below) *adjoint functors* ("adjunctions" between categories, cf. [ML71, §IV]), namely pairs of back-and-forth functors (here: between categories of types)

$$\begin{tikzcd}[column sep=huge]
\mathrm{Type}' \arrow[r, phantom, "\perp"] \arrow[r, shift right=2, "\underset{\textbf{right adjoint}}{R}"'] & \mathrm{Type} \arrow[l, shift right=2, "\overset{\textbf{left adjoint}}{L}"'] \arrow[loop right, "R \circ L =: \mathcal{E}"] & \textbf{induced monad}
\end{tikzcd} \tag{25}$$

equipped with a natural *hom-isomorphism* (forming "adjuncts")

$$\mathrm{Hom}_{\mathrm{Type}}\big(-,\, R(-)\big) \xleftrightarrow{\;\widetilde{(-)}\;} \mathrm{Hom}_{\mathrm{Type}'}\big(L(-),\, -\big) \tag{26}$$

and (equivalently), with natural transformations

$$\overset{\textbf{adjunction unit}}{\mathrm{ret}^{RL}_{D} \;\equiv\; \widetilde{\mathrm{id}_{L(D)}}} \;:\; D \longrightarrow R \circ L(D) \qquad\qquad \overset{\textbf{adjunction co-unit}}{\mathrm{obt}^{LR}_{D'} \;\equiv\; \widetilde{\mathrm{id}_{R(D')}}} \;:\; L \circ R(D') \longrightarrow D'$$

---

[13] The terminology "monad" for (22) is due to [Bé67, §5.4], together with the observation that these are equivalently lax 2-functors from the terminal (point) category $*$ to the ambient 2-category (of type universes, in our case), in which 2-category theoretic sense they are quite the "indecomposable units" which the ancient called monads (as in Euclid: *Elements*, Book VII, Defs. 1, 2, 7, 11). For the present purpose, it is useful to envision that programs running *in* (the Kleisli category of) an effect-monad cannot sensibly interact with other programs until they are taken out (the Kleisli category of) the monad by an effect handler (39).

$$\begin{tikzcd}
\overset{\textbf{adjunction unit}}{\Big(D \xrightarrow{\mathrm{ret}^{RL}_D} R \circ L(D)\Big)} & \Big(L(D) \xrightarrow{\mathrm{id}_{L(D)}} L(D)\Big) \arrow[l, mapsto] \\
\Big(R(D') \xrightarrow{\mathrm{id}_{R(D')}} R(D')\Big) \arrow[r, mapsto] & \underset{\textbf{adjunction counit}}{\Big(L \circ R(D') \xrightarrow{\mathrm{obt}^{LR}_{D'}} D'\Big)}
\end{tikzcd}$$

satisfying the *zig-zag identities*

$$\mathrm{obt}^{LR}_{L(D)} \circ L\big(\mathrm{ret}^{RL}_D\big) \;=\; \mathrm{id}_D \qquad\qquad R\big(\mathrm{obt}^{LR}_{D'}\big) \circ \mathrm{ret}^{RL}_{R(D')} \;=\; \mathrm{id}_{D'}\,,$$

from which the monad structure (23) on $\mathcal{E} := R \circ L$ is obtained as:

$$\begin{tikzcd}
D \arrow[r, "\mathrm{ret}^{\mathcal{E}}_D"] \arrow[d, phantom, "\parallel" rotate=90] & \mathcal{E}(D) \arrow[d, phantom, "\parallel" rotate=90] & \mathcal{E}\big(\mathcal{E}(D)\big) \arrow[r, "\mathrm{join}^{\mathcal{E}}_D"] \arrow[d, phantom, "\parallel" rotate=90] & \mathcal{E}(D) \arrow[d, phantom, "\parallel" rotate=90] \\
D \arrow[r, "\mathrm{ret}^{RL}_D"] & R \circ L(D) & R \circ \underbrace{L \circ R} \circ L(D) \arrow[r, "R\big(\mathrm{obt}^{RL}_{L(D)}\big)"] & R \circ L\,.
\end{tikzcd} \tag{27}$$

Typing of effects via Strong monads. As a technical aside, beware that in describing effecy monad structure this way means to view only its external action on the category of data types. In contrast, when actually coding monadic side effects in programming language constructs (as in §6 below), then the return- and bind-operations (21) will be typed *not* externally as

$$\mathtt{return}^{\mathcal{E}}_D \;:\; \mathrm{Hom}\big(D,\, \mathcal{E}(D)\big) \qquad \text{and} \qquad \mathtt{bind}^{\mathcal{E}}_{D_1,\,D_2} \;:\; \mathrm{Hom}\big(D_1,\, \mathcal{E}(D_2)\big) \longrightarrow \mathrm{Hom}_{\mathrm{Type}}\big(\mathcal{E}(D_1),\, \mathcal{E}(D_2)\big)$$

but internally as terms of iterated *function type* (cf. [McDU22, Def. 5.6] with [BHM02, §4.1][Mi19, §20.2]):

$$\begin{aligned}
\mathtt{return}^{\mathcal{E}}_D \;:\; D \to \mathcal{E}(D)\,, \qquad\qquad \mathtt{bind}^{\mathcal{E}}_{D_1,\,D_2} \;&:\; \big(D_1 \to \mathcal{E}(D_2)\big) \to \big(\mathcal{E}(D_1) \to \mathcal{E}(D_2)\big) \\
&=\; \mathcal{E}(D_1) \times \big(D_1 \to \mathcal{E}(D_2)\big) \to \mathcal{E}(D_2) \\
&=\; \mathcal{E}(D_1) \to \Big(\big(D_1 \to \mathcal{E}(D_2)\big) \to \mathcal{E}(D_2)\Big)\,,
\end{aligned} \tag{28}$$

where

$$(\text{-}) \to (\text{-}) \;\;\equiv\;\; [\text{-},\,\text{-}] \;\;:\;\; \mathrm{Type}^{\mathrm{op}} \times \mathrm{Type} \longrightarrow \mathrm{Type}$$

denotes the formation of function types interpreted as the internal hom-objects in the monoidal closed category of types (eg. [LS86, §I][Bor94b, §6.1]). (Here we stick to notation for cartesian monoidal structure just for the purpose of exposition, see (56) for the analogous non-classical/linear case.)

With the above monad structure phrased internally this way, it is actually richer/stronger, whence one speaks of *enriched* or equivalently *strong monads* ([Mog91, §3.2], review in [**?**, §3.2][McDU22, Prop. 5.8]), here with respect to the self-enrichment of the monoidal closed category of types.

For monads on genuinely classical types (like sets) the strength/enrichment actually exists uniquely (see [McDU22, Ex. 3.7]), but for cases such as linear types (15) it needs to be established (which we do in Prop. 3.6). A convenient way to obtain/verify this enriched/strong monad structure is via symmetric monoidal monad structure:

When the category of types is *symmetric* monoidal closed ([EK66, §III.6], which is the case we are concerned with throughout, cf. Prop. 3.3), then *symmetric monoidal* structure on a monad $\mathcal{E}$, ie.

$$\begin{tikzcd}
\mathcal{E}(D) \times \mathcal{E}(D') \arrow[d, "\mu^{\mathcal{E}}_{D,D'}"] & & \mathcal{E}(D) \times \mathcal{E}(D') \arrow[r, "\sigma_{\mathcal{E}(D),\mathcal{E}(D')}"] \arrow[d, "\mu^{\mathcal{E}}_{D,D'}"] & \mathcal{E}(D') \times \mathcal{E}(D) \arrow[d, "\mu^{\mathcal{E}}_{D',D}"] & D \otimes D' \arrow[r, "\mathrm{ret}^{\mathcal{E}}_D \times \mathrm{ret}^{\mathcal{E}}_{D'}"] \arrow[d, equal] & \mathcal{E}(D) \times \mathcal{E}(D') \arrow[d, "\mu^{\mathcal{E}}"] \\
\mathcal{E}\big(D \times D'\big) & \text{such that} & \mathcal{E}\big(D \times D'\big) \arrow[r, "\mathcal{E}\left(\sigma_{D,\,D'}\right)"'] & \mathcal{E}\big(D' \times D\big) & D \times D' \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{D \times D'}"] & \mathcal{E}(D \times D')
\end{tikzcd} \quad \text{etc.} \tag{29}$$

bijectively induces "commutative" strong monad structure ([Ko72, Thm. 2.3], detailed review in [GLLN08, §7.3, §A.4] [**?**, Prop. 3.3.9]) hence in particular the required enriched monad structure (28).

Examples of effect monads. Fundamental examples of effect monads in classical computer science (and in their linear version of profound importance to us in §4) include (cf. [Mog91, Ex. 1.1]):

- The **reader monad** (e.g. [Mi19, §21.2.3][Uu21, p. 22])

$$\begin{array}{ccccc} R \times (\text{-}) & : & \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ & & D & \longmapsto & [R,\, D] \end{array} \tag{30}$$

  induced from the canonical *comonoid* structure on any cartesian type (given by its terminal and diagonal map):

$$\begin{tikzcd}
\substack{\textbf{comonoid } R \\ \text{(ambient data)}} & & R \times R & & R \arrow[ll, "\mathrm{diag}_R"'] \arrow[rr, "\exists!"] & & \ast \\
R\textbf{-reader monad} & {[R,\,[R,\,D]]} \arrow[r, phantom, "\simeq"] & {[R{\times}W,\,D]} \arrow[rr, "\mathrm{join}_D^{W\mathrm{Reader}}", "{\equiv [\mathrm{diag}_R,\, D]}"'] & & {[R,\,D]} & & {[\ast,\,D] \simeq D} \arrow[ll, "\mathrm{ret}_D^{W\mathrm{Reader}}"', "\equiv \mathrm{const}"]
\end{tikzcd} \tag{31}$$

  Hence a R-Reader-effectful program is one whose nominal output is indefinite until a global parameter $r : R$ is read in, and the handling of R-Reader-effects is the handing-along of this global parameter.

- The **writer monad** (e.g. [Mi19, §4.1 & §21.2.4][Uu21, 1, p. 23]):

$$\begin{array}{ccccc} W \times (\text{-}) & : & \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ & & D & \mapsto & W \times D\,. \end{array} \tag{32}$$

  induced from any *monoid* (aka *unital semi-group*) structure on a type W,

$$\begin{tikzcd}
\substack{\textbf{monoid } W \\ \text{(data output stream)}} & W \times W \arrow[r, "\mathrm{prod}_W"] & A & \ast \arrow[l, "\mathrm{unit}_W"'] \\
A\textbf{-writer monad} & W \times W \times D \arrow[r, "\mathrm{join}_F^{W\mathrm{Writer}} \equiv", "\mathrm{prod}_W \times \mathrm{id}_D"'] & W \times D & \ast \times D = D \arrow[l, "\mathrm{ret}_D^{A\mathrm{Writer}} \equiv"', "\mathrm{unit}_A \times \mathrm{id}_D"]
\end{tikzcd} \tag{33}$$

  (Here the unitality and associativity properties of the monoid W are evidently equivalent to the corresponding properties (24) of the associated writer monad.) In typical applications W is a *free monoid* on an alphabet, hence is the type of *strings* of such characters with multiplication given by concatenation of strings.

  Therefore a Writer-effectful program is one which in addition to its nominal output produces a string (a log message), and the binding of cumulative such effects is by concatenating these strings (appending these messages to the log).

- The **state monad** (e.g. [PP02, §3][Mi19, §21.2.5][Uu21, 1, p. 24])

$$\begin{array}{ccccc} \big[W,\, W \times (\text{-})\big] & : \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ & D & \mapsto & \big[W,\, W \times D\big] \end{array} \tag{34}$$

  given by

$$\begin{tikzcd}
\Big[W,\; W \times \big[W,\, W \times D\big]\Big] \arrow[r, "\mathrm{join}_D^{W\mathrm{State}}"] & \big[W,\, W \times D\big] & D \arrow[l, "\mathrm{ret}_D^{W\mathrm{State}}"'] \\
f \arrow[r, phantom, "\longmapsto"] & \mathrm{ev}\big(f(\text{-})\big)
\end{tikzcd} \tag{35}$$

  Hence WState-effectful programs are adjoint (26) to programs of the form (11)

$$\big(D \xrightarrow{\;\mathrm{prog}\;} [W,\, W \times D']\big) \quad \leftrightarrow \quad \big(W \times D \xrightarrow{\;\widetilde{\mathrm{prog}}\;} W \times D'\big)$$

  and may be understood as producing its nominal output only after it *reads in* data from "memory" type W (as such like the WReader monad above, but) while also re-setting (re-writing) the W-data that gets handed along to a new state.

  This way the state monad is the basic computational model[14] for a *random access memory* ("RAM", see [Ya19, p. 26 & Fig. 1.10]):

$$\begin{tikzcd}
D \arrow[rr, "W\textbf{-RAM effectful program}"] & & \big[W,\, W {\times} D'\big] & \substack{\textbf{type of} \\ W\textbf{State-effectful } D'\textbf{-data}} \\
d \arrow[rr, phantom, "\longmapsto"] & & \Big(w \mapsto \big(w'_{(w,d)},\; d'_{(w,d)}\big)\Big) \\
\textbf{nominal input data} & & \textbf{RAM readout} \quad \textbf{RAM rewrite} \quad \textbf{nominal output data}
\end{tikzcd} \tag{36}$$

---

[14] For practical purposes, the state monad is only a crude model for RAM, since it only encodes access to the entire memory at once (first read all of memory then re-write all of memory). In practice, one will want to read/write RAM only partially at a given address. This is also encoded by a (co-)monadic construction: "lenses", which are the modales over the dual of the state monad: The co-state co-monad [O'C11].

We see these examples in action, together with their linear version, in§4.

One more example (which is not central to our discussion here but is) illustrative of the general notion of computational side effects is the **throwing of exceptions** (e.g. [Mi19, §21.2.6][Uu21, 1, p. 11]): Assuming that the category Type has coproducts and with $\mathrm{Msg} : \mathrm{Type}$ some type of error messages, the exception monad is

$$\begin{array}{rccc} \mathrm{Exc}_{\mathrm{Msg}} & : & \mathrm{Type} \longrightarrow & \mathrm{Type} \\ & & D \quad \mapsto & D \sqcup \mathrm{Msg} \end{array} \tag{37}$$

whose monad unit is the coprojection of the coproduct and whose monad multiplication is given by the co-diagonal on Msg: An $\mathrm{Exc}_{\mathrm{Msg}}$-effectful program with nominal output type D_2 is a morphism $D_1 \longrightarrow D_2 \sqcup \mathrm{Msg}$ which *may* return output of type D_2 but might instead produce an (error-message) term of type Msg, in which case all subsequently $\mathrm{Exc}_{\mathrm{Msg}}$-bound programs will not execute but just hand this error message along. (Hence for $\mathrm{Msg} \equiv *$ the singleton type, this is also known as the *maybe monad.*)

In this example, it is clear that one will wish for programs that can *handle* the exception, and hence in general for programs that can handle a given type of side-effect.

Effect handling and modal types. Given a type of computational side effect $\mathcal{E}$ as above, a program of nominal input type D_1 which can *handle* the effect will have actual input type $\mathcal{E}(D_1)$, and handle the effect-part of $\mathcal{E}(D)$ in a way compatible with the incremental binding of effects:

$$\begin{tikzcd}
 & D_1 \arrow[r, "\mathrm{prog}_{12}", "\text{in-effectful program}"'] & D_2 \arrow[d, bend left, "\text{incorporate handling of } \mathcal{E}(-)\text{-effects}"] \\
 & \mathcal{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathcal{E}}_{D_2}\mathrm{prog}_{12}", "\text{in-effectful program handling effects of type } \mathcal{E}(-)"'] & D_2 \\
D_1 \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D_1}", "\text{produce trivial effect}"'] \arrow[rr, bend right, "\mathrm{prog}_{12}"', "\text{no effect}"'] & \mathcal{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathcal{E}}_{D_2}\mathrm{prog}_{12}", "\text{handle effects running program}"'] & D_2 \\
\mathcal{E}(D_0) \arrow[r, "\mathtt{bind}^{\mathcal{E}}_{\mathrm{prog}_{01}}", "\text{carry effects along}"'] \arrow[rr, bend right, "\mathrm{hndl}^{\mathcal{E}}_{D_2}\big(D_0 \xrightarrow{\mathrm{prog}_{01}} \mathcal{E}(D_1) \xrightarrow{\mathrm{hndl}^{\mathcal{E}}_{D_2}\mathrm{prog}_{12}} D_2\big)"', "\text{handle effects... consecutively}"'] & \mathcal{E}(D_1) \arrow[r, "\mathrm{hndl}^{\mathcal{E}}_{D_2}\mathrm{prog}_{12}", "\text{handle cumulative effects}"'] & D_2
\end{tikzcd} \tag{38}$$

consistency conditions

Such $\mathcal{E}$-effect handling structure on a type D is equivalent to $\mathcal{E}$-**modale**-structure on D (also known as an $\mathcal{E}$-*module* or $\mathcal{E}$-*algebra* structure), namely a morphism

$$\begin{tikzcd} \mathcal{E}(D) \arrow[r, "\rho \equiv \mathrm{hndl}^{\mathcal{E}}_{D}\mathrm{id}_D", "\text{monad action on modale}"'] & D \end{tikzcd} \tag{39}$$

satisfying the axioms of a monoid action, in that it makes the following squares commute:

$$\text{unitality} \qquad \begin{tikzcd} D \arrow[d, "\eta_D"'] \arrow[dr, bend left, "\mathrm{id}"] \arrow[dr, phantom, "\mathrm{utl}_{\mathcal{E}}(\rho)"] & \\ \mathcal{E}(D) \arrow[r, "\rho"'] & D \end{tikzcd} \qquad \text{action property} \qquad \begin{tikzcd} \mathcal{E}\big(\mathcal{E}(D)\big) \arrow[r, "\mathcal{E}(\rho)"] \arrow[d, "\mu_D"] \arrow[dr, phantom, "\mathrm{act}_{\mathcal{E}}(\rho)"] & \mathcal{E}(D) \arrow[d, "\rho"] \\ \mathcal{E}(D) \arrow[r, "\rho"'] & D \end{tikzcd} \tag{40}$$

Categories of effect-handling types A *homomorphism* $(D_1, \rho_1) \to (D_2, \rho_2)$ of $\mathcal{E}$-effect handlers, hence of $\mathcal{E}$-modales, is a map of the underlying data types $f : D_1 \longrightarrow D_2$ which respects the $\mathcal{E}$-action in that the following diagram commutes

$$\begin{tikzcd} \mathcal{E}(D_1) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\rho_1"] & \mathcal{E}(D_2) \arrow[d, "\rho_2"] \\ D_1 \arrow[r, "f"] & D_2 \end{tikzcd}$$

This makes a **category of $\mathcal{E}$-modales** (traditionally known as the *Eilenberg-Moore category* of $\mathcal{E}$ and) traditionally denoted by super-scripting: $\mathrm{Type}^{\mathcal{E}}$.

For example, for any $B : \mathrm{Type}$, the type $\mathcal{E}(B)$ carries $\mathcal{E}$-modale structure, with $\rho \equiv \mu_B$. These are called the *free* $\mathcal{E}$-modales and the full sub-category they form is traditionally denoted by sub-scripting, $\mathrm{Type}_{\mathcal{E}}$:

$$\begin{tikzcd}[column sep=huge]
 & \substack{\textbf{free } \mathcal{E}\textbf{-modales in } \mathrm{Type} \\ (\text{“Kleisli category”})} & \substack{\mathcal{E}\textbf{-modales in } \mathrm{Type} \\ (\text{“Eilenberg-Moore category”})} \\
\mathrm{Type} \arrow[r, two heads, "\overset{\textbf{free construction}}{F_{\mathcal{E}}}"] \arrow[rr, bend right, "F^{\mathcal{E}}"] & \mathrm{Type}_{\mathcal{E}} \arrow[r, hook, "\overset{\textbf{total comparison functor}}{K_{U^{\mathcal{E}}F^{\mathcal{E}}}}"] & \mathrm{Type}^{\mathcal{E}} \\
\big\{ B : \mathrm{Type} \big\} & \big\{ \mathcal{E}(B),\, \rho_B \equiv \mu_B : \mathcal{E}\big(\mathcal{E}(B)\big) \to \mathcal{E}(B) \big\} & \big\{ D : \mathrm{Type},\, \rho : \mathcal{E}(D) \to D \,\big|\, \mathrm{untl}_{\mathcal{E}}(\rho), \mathrm{act}_{\mathcal{E}}(\rho) \big\}
\end{tikzcd} \tag{41}$$

Concretely, the *Kleisli equivalence* re-identifies the homomorphism of free $\mathcal{E}$-modales with the $\mathcal{E}$-effectful programs that we started with (21), as follows (e.g. [Bor94b, Prop. 1.4.6]):

$$\begin{tikzcd}
\mathrm{Type}_{\mathcal{E}} \arrow[rr, hook] & & \mathrm{Type}^{\mathcal{E}} \\
D & \longmapsto & \big(\mathcal{E}(D),\, \mu_D\big) \\
\mathrm{Type}_{\mathcal{E}}(D, D') \arrow[rr, leftrightarrow, "\sim"] & & \mathrm{Type}^{\mathcal{E}}\Big(\big(\mathcal{E}(D), \mu_D\big),\, \big(\mathcal{E}(D'), \mu_{D'}\big)\Big) \\
\Big(D \xrightarrow{\quad f \quad} \mathcal{E}(D')\Big) & \mapsto & \Big(\mathcal{E}(D) \xrightarrow{\mathcal{E}(f)} \mathcal{E}\big(\mathcal{E}(D')\big) \xrightarrow{\mu_{D'}} \mathcal{E}(D')\Big) \\
\Big(D \xrightarrow{\mathrm{ret}^{\mathcal{E}}_D} \mathcal{E}(D) \xrightarrow{\phi} \mathcal{E}(D')\Big) & \mapsfrom & \Big(\mathcal{E}(D) \xrightarrow{\qquad \phi \qquad} \mathcal{E}(D')\Big).
\end{tikzcd} \tag{42}$$

This free construction is readily checked to be left adjoint to evident forgetful functors

$$\begin{tikzcd}[column sep=huge]
\mathrm{Type}_{\mathcal{E}} \arrow[r, hook, "K_{U^{\mathcal{E}}F^{\mathcal{E}}}" description] \arrow[rr, bend left, "U_{\mathcal{E}}"] & \mathrm{Type}^{\mathcal{E}} \arrow[r, "U^{\mathcal{E}}", "\textbf{forgetful functor}"'] & \mathrm{Type} \\
 & \big(D, \rho : \mathcal{E}(D) \to D\big) \arrow[r, mapsto] & D
\end{tikzcd} \tag{43}$$

and both adjunctions $F_{\mathcal{E}} \dashv U_{\mathcal{E}}$ and $F^{\mathcal{E}} \dashv U^{\mathcal{E}}$ re-induce (25) the original monad, with the modale structure recovered from the adjunction counit obt (e.g. [ML71, §VI.2, Thm. 1, §IV.5, Thm. 1]):

$$\begin{tikzcd}
 & & U^{\mathcal{E}}F^{\mathcal{E}}U^{\mathcal{E}}(D, \rho) \arrow[r, equal] \arrow[d, "U^{\mathcal{E}}(\mathrm{obt}_{(D,\rho)})"'] & \mathcal{E}(D) \arrow[d, "\rho"] \\
(D, \rho) \;:\; \mathrm{Type}^{\mathcal{E}} & \vdash & U^{\mathcal{E}}(D, \rho) \arrow[r, equal] & D
\end{tikzcd} \tag{44}$$

In fact, *every* adjunction which induces $\mathcal{E}$ is "in between" these two adjunctions, in that it fits into a commuting diagram of the following form (e.g. [ML71, §VI.3]):

$$\begin{tikzcd}[column sep=huge, row sep=large]
 & & \mathrm{Type}_{\mathcal{E}} \arrow[d, hook, "K_{UF}", "\substack{\textbf{initial} \\ \textbf{comparison} \\ \textbf{functor}}"'] \arrow[dll, bend left=10, "U_{\mathcal{E}}"] & \substack{\textbf{free } \mathcal{E}\textbf{-modales in } \mathrm{Type} \\ (\text{“Kleisli category”})} \\
\textbf{induced monad} \;\; \mathcal{E} \circlearrowright \mathrm{Type} \arrow[urr, two heads, bend left=10, "B \mapsto \left(\mathcal{E}(B),\, \rho \equiv \mu_B\right)", "F_{\mathcal{E}}"'] \arrow[urr, phantom, "\perp"] \arrow[rr, shift left, "F"] \arrow[rr, phantom, "\perp"] \arrow[drr, bend left=10, "F^{\mathcal{E}}"] \arrow[drr, phantom, "\perp"] & & \mathrm{Type}' \arrow[ll, shift left, "U"] \arrow[d, "K^{UF}", "\substack{\textbf{terminal} \\ \textbf{comparison} \\ \textbf{functor}}"'] & \textbf{any adjunction} \text{ for } \mathcal{E} \\
 & & \mathrm{Type}^{\mathcal{E}} \arrow[ull, bend left=10, "U^{\mathcal{E}}", "\textbf{monadic adjunction}"'] & \substack{\mathcal{E}\textbf{-modales in } \mathrm{Type} \\ (\text{“EM-category”})}
\end{tikzcd} \tag{45}$$

The monadicity theorem (cf. [Bor94b, Thm. 4.4.4]) characterizes the monadic adjunctions on the bottom of diagram (45): For a functor U to be *monadic* in that it is of the form $U^{\mathcal{E}}$ in (45), it is sufficient[15] that

---

[15] The necessity clause involves the preservation of "split coequalizers" which we disregard here for brevity since we will not need it.

(i) U is conservative (reflects isomorphisms),
(ii) U has a left adjoint F,
(iii) $\mathrm{dom}(U)$ has coequalizers and U preserves them;

and hence for a functor U between cocomplete categories monadic it is, in particular, sufficient that:

(i) U is conservative,
(ii) U has besides the left adjoint F also a right adjoint,

in which case:

$$\Rightarrow \quad \begin{tikzcd} \mathcal{D} \arrow[d, shift left=2, "U"] \arrow[d, phantom, "\dashv"] \\ \mathrm{Type} \arrow[u, shift left=2, "F"] \end{tikzcd} \text{ is monadic} \quad \Rightarrow \quad \begin{tikzcd} \mathcal{D} \arrow[rr, "\sim"] \arrow[dr, shift left=2, "U"] \arrow[dr, phantom, "\dashv"] & & \mathrm{Type}^{\mathcal{E}} \arrow[dl, shift left=2, "U^{\mathcal{E}}"] \arrow[dl, phantom, "\dashv"] \\ & \mathrm{Type} \arrow[ul, shift left=2, "F"] \arrow[ur, shift left=2, "F^{\mathcal{E}}"] \arrow[loop below, "\mathcal{E}"] & \end{tikzcd} \tag{46}$$

Computational contexts and co-monads on the type system. All of this discussion has a formally dual incarnation (by reversal of all arrows in the above diagrams), now given by *co-monads* on the type system, which some authors refer to as "computational co-effect" but which may naturally be understood as expressing *computational contexts* [?][?]. The idea now is, dually, that a program which *nominally reads in* data of some type D while however executing in dependence on some further context must *de facto* read in data of some adjusted type $\mathcal{C}(D)$ which is such that the context-part of the adjusted data is being transferred to followup programs:

$$\begin{tikzcd}[column sep=large] \mathcal{C}(D_1) \arrow[r, "\overset{\text{first program}}{\mathrm{prog}_{12}}", "\substack{\text{output data} \\ \text{of type } D_2 \\ \text{obtained in context of type } \mathcal{C}(-)}"'] & D_2 & \mathcal{C}(D_2) \arrow[r, "\overset{\text{second program}}{\mathrm{prog}_{23}}", "\substack{\text{input data} \\ \text{of nominal type } D_2 \\ \text{having context of type } \mathcal{C}(-)}"'] & D_3 \arrow[d, mapsto, bend left, "\substack{\text{extend previous context} \\ \text{over second program}}"] \\ \mathcal{C}(D_1) \arrow[r, "\mathtt{extd}^{\mathcal{E}}{}_{\mathrm{prog}_{12}}", "\substack{\text{extend any previous} \\ \mathcal{C}(-)\text{-context going forward}}"'] & \mathcal{C}(D_2) & \mathcal{C}(D_2) \arrow[r, "\mathrm{prog}_{23}"] & D_3 \arrow[d, mapsto, bend left, "\text{compose}"] \\ \mathcal{C}(D_1) \arrow[rrr, "\overset{\mathcal{C}\text{-composite program}}{\mathrm{prog}_{23} \circ \mathtt{extd}^{\mathcal{C}}{}_{\mathrm{prog}_{12}}}", "\text{in shared } \mathcal{C}(-)\text{-context}"'] & & & D_3 \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{C}(D) \arrow[r, "\mathtt{obtain}^{\mathcal{C}}_{D}"] & D \end{tikzcd}$$

obtain plain data from $\mathcal{C}(-)$-context

$$\begin{tikzcd} \mathcal{E}(D) \arrow[r, "\mathtt{extd}^{\mathcal{C}}{}_{\mathtt{obtain}^{\mathcal{C}}_{D}}", "= \mathrm{id}_{\mathcal{E}(D)}"'] & \mathcal{E}(D) \end{tikzcd}$$

Further, by formal duality, all the above discussion for monadic effects and their modal types gives rise to analogous phenomena of comonadic contexts and their (co)modal types. In particular, comonads are induced on the other sides of an adjunction (25):

$$\begin{tikzcd}[column sep=huge] \mathrm{Type}' \arrow[r, shift right=2, "R", "\text{right adjoint}"'] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift right=2, "\overset{\text{right adjoint}}{L}"'] \arrow[loop right, "L \circ R =: \mathcal{C} \quad \text{induced co-monad}"] \end{tikzcd} \tag{47}$$

Dualizing the previous examples (33)(32) of read/write-effect monads this way, one obtains the following list of **examples** of reader/writer (co)monads:

(Co)monad name	**Underlying endofunctor**	**(Co)monad structure induced by**
Reader monad	[W, -] on cartesian types	unique comonoid structure on W
CoReader comonad	$W \times$(-) on cartesian types	unique comonoid structure on W
Writer monad	$A \otimes$(-) on monoidal types	chosen monoid structure on A
CoWriter comonad	$[A,$ -] on monoidal types $A \otimes$(-) on monoidal types	chosen monoid structure on A chosen comonoid structure on A
Writer/CoWriter Frobenius monad	$A \otimes$(-) on monoidal types	chosen Frob. monoid structure on A

(48)

Adjoint (co)monads. In the case of an *adjoint triple* of adjoint functors the induced (co)monads are themselves pairwise adjoint — as in (4), a situation central to our discussion in §4. In this case their categories of (co)modales

(41) are isomorphic (e.g. [MLM92, §V.8, Thm. 2]):

$$\begin{tikzcd}
\text{adjoint (co)monads} & \text{have} & & \text{equivalent categories of modales} & \\
\mathcal{E} \dashv \mathcal{C} & \vdash & \mathrm{Type}^{\mathcal{E}} \arrow[rr, leftrightarrow, "\sim"] \arrow[dr, "U^{\mathcal{E}}"'] & & \mathrm{Type}^{\mathcal{C}} \arrow[dl, "U^{\mathcal{C}}"] \\
& & & \mathrm{Type} &
\end{tikzcd} \tag{49}$$

Frobenius monads. Something special happens here when the underlying endo-functors in (49) are not just adjoints but also identified, $\mathcal{E} \simeq \mathcal{C}$. In this case their (co)monad structures fuse to a single *Frobenius monad*-structure [Law69b, pp. 151][Str04][Lau06] — induced via (45) and (47) from an "ambidextrous" adjunction, where the left and the right adjoint of a middle functor agree (5) — so-called because these monads are *Frobenius algebras* (Frobenius monoids, see e.g. [HV19, §5]) internal to the category of endofunctors: Combined (co)algebras whose (co)products are compatible in the sense that all ways that map n input elements to m output elements by $(n-1)$ products and $(m-1)$-coproducts coincide.

For **example** – shown in the last line of (47): if type A carries Frobenius algebra structure, then the induced (Co)Reader (co)monad $A \otimes (\text{-})$ carries induced Frobenius monad structure.

Literature 2.16 (**Classical structures via Frobenius monads).** The (co)monad expressing quantum measurement effects which we derive in ... was originally proposed in ... [CPav08][CPac08][CPP0909][CPV12], partial review in [HV19]. Its graphical formalization as part of the `zxCalculus`[16] (review in: [vWe][Co23]) originates in [CD08, §3][CD11, Def. 6.4][Ki08, §§2][Ki09, §4] (...)

---

[16] `zxCalculus`landing page: `zxcalculus.com`

3 Quantum types

We discuss here (mostly the categorical semantics of) the 0-fragment (Rem. 2.11) of the type system of `LHoTT`, (semantics for the full un-truncated fragment is discussed in [EoS]). This is essentially the model of Proto-`Quipper` from [RS18] (Lit. 2.5), but we present a novel modal/monadic perspective that lends itself to the modal typing of quantum effects in §4 and then to the formulation of the quantum certification language `QS` in §6. Nonetheless, a key point is that Proto-`Quipper`-programs may be translated to `LHoTT`, such as to formally certify them, see [Ri23].

§3.1 – Semantics
§3.2 – Syntax

Linear/Quantum Data Types

Characteristic Property	**1.** Their cartesian product blends into the co-product:	**2.** A tensor product appears & distributes over direct sum	**3.** A linear function type appears adjoint to tensor
Symbol	$\bigoplus$ direct sum	$\otimes$ tensor product	$\multimap$ linear function type
Formula (for B : FinType)	$\overset{\text{cart. product}}{\prod_B \mathcal{H}_b} \simeq \underset{\text{direct sum}}{\bigoplus_B \mathcal{H}_b} \simeq \overset{\text{co-product}}{\coprod_B \mathcal{H}_b}$	$\mathscr{V} \otimes \Big(\bigoplus\limits_{b:B} \mathcal{H}_b \Big) \simeq \bigoplus\limits_{b:B} (\mathscr{V} \otimes \mathcal{H}_b)$	$(\mathscr{V} \otimes \mathcal{H}) \multimap \mathscr{K}$ $\simeq \mathscr{V} \multimap (\mathcal{H} \multimap \mathscr{K})$
AlgTop Jargon	biproduct, stability, ambidexterity	Frobenius reciprocity	mapping spectrum
		Grothendieck's Motivic Yoga of 6 oper.	(Wirthmüller form)
Linear Logic	additive disjunction	multiplicative conjunction	linear implication
Physics Meaning	parallel quantum systems	compound quantum systems	qRAM systems

3.1 Quantum Semantics

We start with a concrete example (Def. 3.1 below) of a category which interprets (as we shall see in §3.2) a small fragment of `LHoTT` relevant for expressing quantum circuits (in **??**). Category-theoretically this example is elementary and standard (going back to [Bé85, §3.3][HT95, pp. 281]), but it is important in applications, eg. as the established model for Proto-`Quipper` (Lit. 2.5, where it appears as [RS18, Def. 3.3] for the case that their fiber category $\overline{M}$ is the category $\mathrm{Mod}_{\mathbb{C}}$ of complex vector bundles). Here we highlight previously underappreciated aspects of this model (all shared by its homotopy-theoretic generalizations in [EoS]):

- its doubly closed monoidal structure (Prop. 3.3),
- its doubly strong monadic reflections (Prop. 3.6)
- its quantization/exponential modality (Prop. 3.8)
- its support of 6-operations motivic yoga (Prop. 3.14)

which make the model interpret an expressive modal/monadic/effectful quantum language `QS`, in §6.

Last but not least, the model serves to transparently elucidate key language features of `LHoTT` in §3.2.

Definition 3.1 (**Category of linear bundle types**).
For the purpose of this section, we write "Type" for the category equivalently described as follows
(cf. [EoS], where this category is denoted "$\mathrm{Fam}_{\mathbb{C}}$"):

- Type is the free coproduct completion of $\mathrm{Mod}_{\mathbb{C}}$,
- Type is the category of *indexed sets* of complex vector spaces,
- Type is the category of complex vector bundles over varying discrete base spaces,
- Type is the 0-sector of the ∞-category of ∞-local systems over varying general base spaces,
- Type is the Grothendieck construction of the Set-indexed category whose fiber over W : Set is the category $\mathrm{Mod}_{\mathbb{C}}^{W} \equiv \mathrm{Func}(W, \mathrm{Mod}_{\mathbb{C}})$ of W-indexed complex vector spaces (complex vector bundles over W):

Types	**Category**	**Morphisms**
ClaType classical types	Set sets	$W \xrightarrow{\quad f \quad} W'$ maps
QuType linear types	$\mathrm{Mod}_{\mathbb{C}}$ vector spaces	$\mathcal{H} \xrightarrow{\quad \phi \quad} \mathcal{H}'$ linear maps
QuType_{W} W-dependent linear types	$\mathrm{Mod}_{\mathbb{C}}^{W}$ W-indexed vector space	$\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \begin{matrix}\xrightarrow{\phi_\bullet} \\ \\ =\!=\!=\end{matrix} \begin{bmatrix}\mathcal{H}'_\bullet \\ \downarrow \\ W\end{bmatrix}.$ W-indexed linear maps
Type linear bundle types	$\int_{W:\mathrm{Set}} \mathrm{Mod}_{\mathbb{C}}^{W}$ Grothendieck construction	$\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \begin{matrix}\xrightarrow{\phi_\bullet} \\ \\ \xrightarrow{f}\end{matrix} \begin{bmatrix}\mathcal{H}'_\bullet \\ \downarrow \\ W'\end{bmatrix}.$ map covered by indexed linear map

(50)

When describing their linear fiber types concretely, we also denote linear bundle types as follows:

$$\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \equiv \begin{bmatrix}\mathcal{H}_w \\ \downarrow \\ (w : W)\end{bmatrix} \equiv (w : W) \times (\mathcal{H}_w : \mathrm{QuType}) \tag{51}$$

Their hom-sets we denote as follows (the second line close to the internal type-theoretic syntax, see Rem. 3.4):

$$\begin{aligned} \mathrm{Hom}\left(\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix}, \begin{bmatrix}\mathcal{H}'_\bullet \\ \downarrow \\ W'\end{bmatrix}\right) &\simeq \big(f : \mathrm{Hom}(W, W')\big) \times \prod_{w} \mathrm{Hom}\big(\mathcal{H}_w, \mathcal{H}'_{f(w)}\big) \\ &\equiv \big(f : W \to W'\big) \times \prod_{w} \natural\big(\phi_w : \mathcal{H}_w \to \mathcal{H}'_{f(w)}\big)\,. \end{aligned} \tag{52}$$

pair types $\mathrm{Hom}\big(X \cdot X',\, X''\big) \;\simeq$	**function types** $\mathrm{Hom}\big(X,\, [X', X'']\big)$
$W \times W'$ cartesian product	$W' \to W''$ set of maps
$\bigoplus_S \mathcal{H}'$ direct sum	$\natural(\mathcal{H}' \to \mathcal{H}'')$ set of linear maps
$\mathcal{H} \otimes \mathcal{H}'$ tensor product	$\mathcal{H} \multimap \mathcal{H}'$ vector space of linear maps
$\bigoplus_S \mathcal{H}'_\bullet$ direct sum	$\prod_w (\mathcal{H}'_w \to \mathcal{H}''_w)$ set of indexed linear maps
$\mathcal{H} \otimes \mathcal{H}'_\bullet$ index-wise tensor product	$\prod_w (\mathcal{H}'_w \multimap \mathcal{H}''_w)$ vector space of indexed linear maps
$\begin{bmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{bmatrix} \times \begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{bmatrix}$ $= \begin{bmatrix} \mathcal{H}_\bullet \oplus \mathcal{H}'_\bullet \\ \downarrow \\ W \times W' \end{bmatrix}$ external direct sum	$\begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{bmatrix} \to \begin{bmatrix} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{bmatrix} =$ $\begin{bmatrix} \prod_{w'} \mathcal{H}''_{f(w')} \\ \downarrow \\ \big(f : W' \to W''\big) \times \\ \prod\limits_{w'} \natural\big(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big) \end{bmatrix}$
$\begin{bmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{bmatrix} \otimes \begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{bmatrix}$ $= \begin{bmatrix} \mathcal{H}_\bullet \otimes \mathcal{H}'_\bullet \\ \downarrow \\ W \times W' \end{bmatrix}$ external tensor product	$\begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{bmatrix} \multimap \begin{bmatrix} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{bmatrix} =$ $\begin{bmatrix} \prod_{w'} \big(\mathcal{H}_{w'} \multimap \mathcal{H}''_{f(w')}\big) \\ \downarrow \\ \big(f : W' \to W''\big) \end{bmatrix}$

Closed monoidal structures on bundle types. First recall:

- ClaType is cartesian closed monoidal, with:
  - monoidal product the Cartesian product ×
  - internal hom the function sets $W \to W'$
  - unit object $*$ the singleton set
- QuType is non-Cartesian closed monoidal with:
  - monoidal product the usual tensor product,
  - internal hom the linear hom-spaces $\mathcal{H} \multimap \mathcal{H}'$
  - unit object the ground field $\mathbb{1} \equiv \mathbb{C} : \mathrm{Mod}_{\mathbb{C}}$.

Remark 3.2 (**External monoidal structures).** Given any monoidal category $(\mathcal{C}, \otimes, \mathbb{1})$, its free coproduct completion $\mathrm{Fam}_{\mathcal{C}}$ (of indexed sets of $\mathcal{C}$-objects) inherits a corresponding "*external*" monoidal struture given by joint fiberwise product in $\mathcal{C}$ over the Cartesian product of index sets (for pointers see [EoS, p. 4]).

Proposition 3.3 (**Doubly closed monoidal structure of linear bundle types).** *The category* Type (50) *of linear bundle types is* doubly closed monoidal*, as shown on the right, in that:*

- *it is cartesian closed with respect to the* external direct sum*,*

  with unit object $* \;\equiv\; \begin{bmatrix} 0 \\ \downarrow \\ * \end{bmatrix} \;:\; \mathrm{Type}$
- *it is non-cartesian closed symmetric monoidal with respect to the* external tensor product (cf. [RS18, Prop. 3.5])

  with unit object $\mathbb{1} \;\equiv\; \begin{bmatrix} \mathbb{1} \\ \downarrow \\ * \end{bmatrix} \;:\; \mathrm{Type}$

Remark 3.4 (Notation for internal homs)**.** The arrow-notation for the hom-sets in QuType and QuType_W is that inherited from Type under the embeddings ClaType, QuType ↪ Type (57), in that:

$$\natural\left(\begin{bmatrix} \mathcal{H} \\ \downarrow \\ * \end{bmatrix} \to \begin{bmatrix} \mathcal{H}' \\ \downarrow \\ * \end{bmatrix}\right) \;=\; \natural(\mathcal{H} \to \mathcal{H}')$$

$$\left(\begin{bmatrix} \mathcal{H} \\ \downarrow \\ * \end{bmatrix} \multimap \begin{bmatrix} \mathcal{H}' \\ \downarrow \\ * \end{bmatrix}\right) \;=\; (\mathcal{H} \multimap \mathcal{H}')$$

where on the right the embeddings (57) are understood.

This way, eg. the natural hom-isomorphism expressing the closed monoidal structure on QuType reads

$$\natural\big(\mathcal{H} \otimes \mathcal{H}' \to \mathcal{H}''\big) \;\simeq\; \natural\big(\mathcal{H} \to (\mathcal{H}' \multimap \mathcal{H}'')\big) \tag{53}$$

But we now also have mixed classical/quantum expressions, notably this one, which is going to be important:

$$\boxed{(W \to \mathcal{H}) \;\equiv\; \begin{bmatrix} 0 \\ \downarrow \\ W \end{bmatrix} \to \begin{bmatrix} \mathcal{H} \\ \downarrow \\ * \end{bmatrix} \;=\; \begin{bmatrix} \prod_W \mathcal{H} \\ \downarrow \\ * \end{bmatrix} \;=\; \begin{bmatrix} \mathbb{1}_\bullet \\ \downarrow \\ W \end{bmatrix} \multimap \begin{bmatrix} \mathcal{H} \\ \downarrow \\ * \end{bmatrix} \;=\; \big(\mathbb{1} \times W \multimap \mathcal{H}\big)} \tag{54}$$

Proof of Prop. 3.3. Since our classical base category is ClaType $\equiv$ Set, it is clearly sufficient to check the defining hom-isomorphism for the case that $W = *$. In this case we have the following sequences of natural isomorphisms:

$$\begin{aligned}
&\mathrm{Hom}\left(\begin{bmatrix}\mathcal{H} \oplus \mathcal{H}'_\bullet \\ \downarrow \\ W'\end{bmatrix}, \begin{bmatrix}\mathcal{H}''_\bullet \\ \downarrow \\ W''\end{bmatrix}\right) \\
&\simeq\; (f : W' \to W'') \times \prod_{w'} \natural\big(\mathcal{H} \oplus \mathcal{H}'_{w'} \to \mathcal{H}''_{f(f')}\big) && \text{by (52)} \\
&\simeq\; (f : W' \to W'') \times \prod_{w'} \Big(\natural\big(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big) \times \natural\big(\mathcal{H} \to \mathcal{H}''_{f(w')}\big)\Big) && \text{by coproduct property of } \oplus \\
&\simeq\; (f : W' \to W'') \times \prod_{w'} \natural\big(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big) \times \prod_{w'} \natural\big(\mathcal{H} \to \mathcal{H}''_{f(w')}\big) && \text{since } \textstyle\prod_W(-) \text{ is right adjoint} \\
&\simeq\; (f : W' \to W'') \times \prod_{w'} \natural\big(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big) \times \natural\Big(\mathcal{H} \to \prod_{w'} \mathcal{H}''_{f(w')}\Big) && \text{since } \mathcal{H} \to \text{(-) is right adjoint} \\
&\simeq\; \mathrm{Hom}\left(\begin{bmatrix}\mathcal{H} \\ \downarrow \\ *\end{bmatrix}, \begin{bmatrix}\prod_{w'} \mathcal{H}''_{f(w')} \\ \downarrow \\ (f : W' \to W'') \times \prod\limits_{w'} \big(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big)\end{bmatrix}\right) && \text{by (52)}
\end{aligned}$$

and

$$\begin{aligned}
&\mathrm{Hom}\left(\begin{bmatrix}\mathcal{H} \otimes \mathcal{H}'_\bullet \\ \downarrow \\ W'\end{bmatrix}, \begin{bmatrix}\mathcal{H}''_\bullet \\ \downarrow \\ W''\end{bmatrix}\right) \\
&\simeq\; \big(f : W' \to W''\big) \times \prod_{w'} \natural\Big(\mathcal{H} \otimes \mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\Big) && \text{by (52)} \\
&\simeq\; \big(f : W' \to W''\big) \times \prod_{w'} \natural\Big(\mathcal{H} \to \big(\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')}\big)\Big) && \text{by (53)} \\
&\simeq\; \big(f : W' \to W''\big) \times \natural\Big(\mathcal{H} \to \prod_{w'} \big(\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')}\big)\Big) && \text{since } \mathcal{H} \to \text{(-) is right adjoint} \\
&\simeq\; \mathrm{Hom}\left(\begin{bmatrix}\mathcal{H} \\ \downarrow \\ *\end{bmatrix}, \begin{bmatrix}\prod_{w'} \big(\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')}\big) \\ \downarrow \\ \big(f : W' \to W''\big)\end{bmatrix}\right) && \text{by (52)}
\end{aligned}$$

which proves the claim. ☐

Remark 3.5. There is a natural comparison morphism from the cartesian to the tensor product, being the universal fiberwise bilieanr map:

$$\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix}, \begin{bmatrix}\mathcal{H}'_\bullet \\ \downarrow \\ W'\end{bmatrix} \;:\; \mathrm{Type} \qquad \vdash \qquad \begin{tikzcd}[column sep=huge]
\begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \times \begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \arrow[r] & \begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \otimes \begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \\
\begin{bmatrix}\mathcal{H}_w \times \mathcal{H}'_{w'} \\ \downarrow \\ (w : W) \times (w' : W')\end{bmatrix} \arrow[r, shift left=3, "\left|\begin{smallmatrix}\psi \\ \psi'\end{smallmatrix}\right\rangle \mapsto |\psi\rangle \otimes |\psi'\rangle"] \arrow[r, equal, shift right=3] & \begin{bmatrix}\mathcal{H}_w \otimes \mathcal{H}'_{w'} \\ \downarrow \\ (w : W) \times (w' : W')\end{bmatrix}
\end{tikzcd} \tag{55}$$

Quantum type declaration. For transparent distinction between the classical and quantum monoial structures from Prop. 3.3 it is convenient to use, besides the standard notation for

- the classical type declaration in the "empty" context

$$\vdash \qquad w : W\,,$$

which really is type declaration in the context of the cartesian monoidal unit $\ast : \mathrm{ClaType}$

$$\ast \qquad \vdash \qquad w : W\,,$$

also notation for

- a linear (quantum) type declaration

$$\vdash \qquad |\psi\rangle \;{}^{\circ}_{\circ}\; \mathcal{H}\,,$$

to be understood as syntactic sugar for (ordinary) type declaration in the context of the tensor monoidal unit:

$$\mathbb{1} \qquad \vdash \qquad |\psi\rangle : \mathcal{H}\,.$$

This little notational device will be particularly useful when declaring data of type $W \to \mathcal{H}$ (54).

Data	**Declaration**	**Semantics**
Classical	$\begin{array}{cccc} & \vdash & W & : & \mathrm{ClaType} \\ & \vdash & w & : & W \end{array}$	$\left[\begin{array}{c} 0 \\ \downarrow \\ \ast \end{array}\right] \begin{array}{c} \xrightarrow{\;0_w\;} \\ \\ \xrightarrow{\;w\;} \end{array} \left[\begin{array}{c} 0_\bullet \\ \downarrow \\ W \end{array}\right]$
Quantum	$\begin{array}{ccccc} & \vdash & \mathcal{H} & : & \mathrm{QuType} \\ & \vdash & \lvert\psi\rangle & {}^{\circ}_{\circ} & \mathcal{H} \\ & & & \rotatebox{90}{$\equiv$} & \\ \mathbb{1} & \vdash & \lvert\psi\rangle & : & \mathcal{H} \end{array}$	$\left[\begin{array}{c} \mathbb{1} \\ \downarrow \\ \ast \end{array}\right] \begin{array}{c} \xrightarrow{\;\lvert\psi\rangle\;} \\ \\ \xrightarrow{\;\ast\;} \end{array} \left[\begin{array}{c} \mathcal{H} \\ \downarrow \\ \ast \end{array}\right]$
Quantized	$\begin{array}{ccccc} & \vdash & W & : & \mathrm{ClaType} \\ & \vdash & \mathcal{H} & : & \mathrm{QuType} \\ & \vdash & \sum\limits_w \lvert w\rangle & {}^{\circ}_{\circ} & W \to \mathcal{H} \\ & & & \rotatebox{90}{$\equiv$} & \\ \mathbb{1} & \vdash & \sum\limits_w \lvert w\rangle & : & W \to \mathcal{H} \end{array}$	$\left[\begin{array}{c} \mathbb{1} \\ \downarrow \\ \ast \end{array}\right] \begin{array}{c} \xrightarrow{\;\sum_w \lvert w\rangle\;} \\ \\ \xrightarrow{\;\ast\;} \end{array} \left[\begin{array}{c} \prod\limits_W \mathcal{H} \\ \downarrow \\ \ast \end{array}\right]$

(56)

Classical and Quantum Modality.

Proposition 3.6 (**Reflective subcategories of purely classical/quantum modal types).**

The category of Def. 3.1 has monadic (45) *reflective subcategory inclusions as follows:*

$$\begin{tikzcd}
W & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \arrow[l, mapsto] & & \bigoplus\limits_{w}\mathcal{H}_w & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \arrow[l, mapsto] \\
\mathrm{ClaType} \arrow[r, hook, shift right] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift right] \arrow[loop right, "\natural\ \text{classically}"] & & \mathrm{QuType} \arrow[r, hook, shift right] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift right] \arrow[loop right, "\triangleright\ \text{quantumly}"] \\
W \arrow[r, mapsto] & \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right] & & \mathcal{H} \arrow[r, mapsto] & \left[\begin{array}{c}\mathcal{H}\\ \downarrow\\ *\end{array}\right]
\end{tikzcd} \tag{57}$$

Moreover, the induced classical/quantum-modalities are strong monads (28) with respect to the classical/quantum monoidal structures of Prop. 3.3, whence we have `return`- and `bind`-operations (21) as follows, using the type declaration from (56):

classically $\natural$	$\mathtt{return}^{\natural}_{W} \;:\; \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \to \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right]$ $\mathtt{return}^{\natural}_{W} \;\equiv\; \vert\psi_w\rangle \;\mapsto\; 0_w$	$\mathtt{bind}^{\natural} \;:\; \left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \to \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W'\end{array}\right]\right) \to \left(\left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right] \to \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W'\end{array}\right]\right)$ $\mathtt{bind}^{\natural} \;\equiv\; \big(\vert\psi_w\rangle \mapsto 0_{f(w)},\, 0\big) \;\mapsto\; \big(0_w \;\mapsto\; 0_{f(w)},\, 0\big)$
quantumly $\triangleright$	$\mathtt{return}^{\triangleright} \;⦂\; \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \multimap \left[\begin{array}{c}\oplus_w\mathcal{H}_w\\ \downarrow\\ *\end{array}\right]$ $\mathtt{return}^{\triangleright} \;\equiv\; \vert\psi_w\rangle \;\mapsto\; \bigoplus\limits_{w'}\delta^{w'}_{w}\vert\psi_w\rangle$	$\mathtt{bind}^{\triangleright} \;⦂\; \left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \multimap \left[\begin{array}{c}\mathcal{H}'\\ \downarrow\\ *\end{array}\right]\right) \multimap \left(\left[\begin{array}{c}\oplus_w\mathcal{H}_w\\ \downarrow\\ *\end{array}\right] \multimap \left[\begin{array}{c}\mathcal{H}'\\ \downarrow\\ *\end{array}\right]\right)$ $\mathtt{bind}^{\triangleright} \;\equiv\; \big(\vert\psi_w\rangle \mapsto F_w\vert\psi_w\rangle\big) \;\mapsto\; \Big(\bigoplus\limits_{w}\vert\psi_w\rangle \mapsto \sum\limits_{w} F_w\vert\psi_w\rangle\Big)$

(58)

Proof. It is evident that the inclusions are fully faithful and reflective. Formally we may check the required hom-isomorphisms (26) using (52):

$$\begin{array}{ll}
\mathrm{Hom}\left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right], \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W'\end{array}\right]\right) & \mathrm{Hom}\left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right], \left[\begin{array}{c}\mathcal{H}'\\ \downarrow\\ *\end{array}\right]\right) \\
\simeq\; \mathrm{Hom}(W, W') \times \prod\limits_{w}\mathrm{Hom}(\mathcal{H}_w, 0) & \simeq\; \mathrm{Hom}(\mathcal{H}, *) \times \prod\limits_{w}\mathrm{Hom}(\mathcal{H}_w, \mathcal{H}') \\
\simeq\; \mathrm{Hom}(W, W') & \simeq\; \mathrm{Hom}\big(\coprod_w \mathcal{H}_w, \mathcal{H}'\big) \\
\simeq\; \mathrm{Hom}\left(\left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right], \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W'\end{array}\right]\right) & \simeq\; \mathrm{Hom}\left(\left[\begin{array}{c}\oplus_w\mathcal{H}_w\\ \downarrow\\ *\end{array}\right], \left[\begin{array}{c}\mathcal{H}'\\ \downarrow\\ *\end{array}\right]\right)
\end{array}$$

To check monadicity, we invoke the monadicity theorem in the form (46): Since both inclusions are right adjoints and evidently conservative, it is sufficient to observe that they both preserve all coequalizers. For this we can appeal to [EoS, Prop. A.9].

Finally, to check that the induced monads are strong, we may equivalently check that they are monoidal (29): The (strong) monoidal structure on the underlying functors is indicated vertically in the following diagrams. Since the monads are idempotent, it is sufficient to check furthermore that their unit transformations is monoidal, hence that these squares commute, which is immediate in components (58):

$$\begin{tikzcd}
\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \times \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \arrow[r, "\mathrm{ret}^\natural"] \arrow[ddd, equal] & \natural\left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \times \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right]\right) \arrow[d, phantom, "\wr\quad\text{by Prop. 3.3}"] \\
& \natural\left[\begin{array}{c}\mathcal{H}_\bullet \oplus \mathcal{H}'_\bullet\\ \downarrow\\ W \times W'\end{array}\right] \arrow[d, phantom, "\wr\quad\text{by (57)}"] \\
& W \times W' \arrow[d, phantom, "\wr\quad\text{by (57)}"] \\
\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \times \left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \arrow[r, "\mathrm{ret}^\natural \times \mathrm{ret}^\natural"] & \natural\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \times \natural\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right]
\end{tikzcd}$$

$$\begin{tikzcd}
\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \otimes \left[\begin{array}{c}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{array}\right] \arrow[r, "\mathrm{ret}^\triangleright"] \arrow[dddddd, equal] & \triangleright\left(\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \otimes \left[\begin{array}{c}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{array}\right]\right) \arrow[d, phantom, "\rotatebox{90}{$\equiv$}\quad\text{from Prop. 3.3}"] \\
& \triangleright\left[\begin{array}{c}\mathcal{H}_\bullet \otimes \mathcal{H}'_\bullet\\ \downarrow\\ W \times W'\end{array}\right] \arrow[d, phantom, "\wr\quad\text{by (57)}"] \\
& \bigoplus_{(w,w')}\left(\mathcal{H}_w \otimes \mathcal{H}'_{w'}\right) \arrow[d, phantom, "\wr"] \\
& \bigoplus_{w}\bigoplus_{w'}\left(\mathcal{H}_w \otimes \mathcal{H}'_{w'}\right) \arrow[d, phantom, "\wr\quad\text{distributivity}"] \\
& \left(\bigoplus_{w}\mathcal{H}_w\right) \otimes \left(\bigoplus_{w'}\mathcal{H}'_{w'}\right) \arrow[d, phantom, "\wr\quad\text{by (57)}"] \\
& \\
\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \otimes \left[\begin{array}{c}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{array}\right] \arrow[r, "\mathrm{ret}^\triangleright \otimes \mathrm{ret}^\triangleright"] & \triangleright\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \otimes \triangleright\left[\begin{array}{c}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{array}\right]
\end{tikzcd}$$

□

Quantum/Classical Data Types		**Quantum/Classical Maps**
general **bundles of** **linear types**	$\begin{tikzcd}\mathrm{Type} \arrow[loop left, "\natural"] \arrow[loop right, "\triangleright"]\end{tikzcd}$ $\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right]$	$\mathcal{H}_\bullet \longrightarrow \mathcal{H}'_\bullet$ $\left[\begin{array}{c}\mathcal{H}_\bullet\\ \downarrow\\ W\end{array}\right] \begin{array}{c}\xrightarrow{\phi_\bullet}\\ =\end{array} \left[\begin{array}{c}\mathcal{H}'_{f(\bullet)}\\ \downarrow\\ W\end{array}\right] \begin{array}{c}\to\\ \xrightarrow{f}\end{array} \left[\begin{array}{c}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{array}\right]$
purely **classical types** (bundles of zeros)	$\mathrm{ClaType} \;\equiv\; \mathrm{Type}^\natural$ $\left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right]$	$W \longrightarrow W'$ $\left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right] \begin{array}{c}\xrightarrow{\;0_\bullet\;}\\ \xrightarrow{\;f\;}\end{array} \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W'\end{array}\right]$
purely **linear types** (bundles over point)	$\mathrm{QuType} \;\equiv\; \mathrm{Type}^\triangleright$ $\left[\begin{array}{c}\mathcal{H}\\ \downarrow\\ *\end{array}\right]$	$\mathcal{H} \longrightarrow \mathcal{H}'$ $\left[\begin{array}{c}\mathcal{H}\\ \downarrow\\ *\end{array}\right] \begin{array}{c}\xrightarrow{\;\phi\;}\\ \longrightarrow\end{array} \left[\begin{array}{c}\mathcal{H}'\\ \downarrow\\ *\end{array}\right]$

In fact, the purely classical types are also coreflective, whence the classical-modality $\natural$ is in fact a *bireflective Frobenius modality* (cf. §3.2.1):

Proposition 3.7.

$$\begin{tikzcd}
\mathrm{ClaType} \arrow[r, hook] & \mathrm{Type} \arrow[l, bend right, "\natural"'] \arrow[l, bend left, "\natural"]
\end{tikzcd} \tag{59}$$

Quantization and Exponential modality. Composing the Cartesian hom-adjunction for $\mathbb{1}$ (from Prop. 3.3) with the classicality-coreflection (59) gives another adjunction between linear bundle types and purely classical types:

$$\begin{tikzcd}
W \arrow[rr, mapsto] & & \left[\begin{smallmatrix} \mathbb{1}_\bullet \\ \downarrow \\ W \end{smallmatrix}\right] \\
\mathrm{ClaType} \arrow[r, hook, shift left=2] \arrow[rr, bend left=40] \arrow[r, phantom, "\perp"] & \mathit{Types} \arrow[l, shift left=2, "\natural"] \arrow[r, shift left=2, "\mathbb{1}\times(\text{-})"] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift left=2, "\mathbb{1}\to(\text{-})"] \arrow[ll, bend left=40] \\
(w : W) \times \natural(\mathbb{1} \to \mathcal{H}_w) & & \left[\begin{smallmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{smallmatrix}\right] \arrow[ll, mapsto]
\end{tikzcd} \tag{60}$$

Further composing (60) with the reflection of purely quantum types (57), gives:

Proposition 3.8 (**Quantization and Classicization**). *We have a pair of adjoint functors between purely classical and purely quantum types* (57) *of this form*

$$\begin{tikzcd}
W \arrow[rr, mapsto] & & \underset{W}{\bigoplus} \mathbb{1} \\
\mathrm{ClaType} \arrow[r, shift left=2, "\mathbb{1}\times(\text{-})"] \arrow[r, phantom, "\perp"] \arrow[rr, bend left=40, "\overset{\text{quantization}}{\mathrm{Q}} \;\equiv\; \overset{\text{motive}}{\Sigma^\infty}"] & \mathrm{Type} \arrow[l, shift left=2, "\natural(\mathbb{1}\to(\text{-}))"] \arrow[r, shift left=2, "\triangleright"] \arrow[r, phantom, "\perp"] & \mathrm{QuType} \arrow[l, hook, shift left=2] \arrow[ll, bend left=40, "\underset{\text{classicized}}{\mathrm{C}} \;\equiv\; \Omega^\infty"] \arrow[loop right, "\boldsymbol{!}", "\text{exponential modality}"'] \\
\natural(\mathbb{1} \to \mathcal{H}) & & \mathcal{H} \arrow[ll, mapsto]
\end{tikzcd} \tag{61}$$

where the composite $! \equiv QC$ *is the "exponential modality"* (Rem. 3.9). *These are monoidal with respect to the classical/quantum monoidal structures (Prop. 3.3) via natural transformations of the following form:*

$$\begin{aligned}
W, W' \,:\, \mathrm{ClaType} \quad &\vdash \quad (\mathrm{Q}W) \otimes (\mathrm{Q}W') \,\simeq\, \mathrm{Q}(W \times W') \\
\mathcal{H}, \mathcal{H}' \,:\, \mathrm{QuType} \quad &\vdash \quad (\mathrm{C}\,\mathcal{H}\,) \times (\mathrm{C}\,\mathcal{H}'\,) \simeq \mathrm{C}(\,\mathcal{H} \,\times\, \mathcal{H}'\,) \\
\mathcal{H}, \mathcal{H}' \,:\, \mathrm{QuType} \quad &\vdash \quad (\mathrm{C}\,\mathcal{H}\,) \times (\mathrm{C}\,\mathcal{H}'\,) \to \mathrm{C}(\,\mathcal{H} \,\otimes\, \mathcal{H}'\,)
\end{aligned} \tag{62}$$

$$\mathrm{Q}\ast \,\simeq\, \mathbb{1}\,, \qquad \mathrm{C}\,0 \,\simeq\, \mathbb{1}\,, \qquad \mathrm{C}\mathbb{1} \,\to\, \mathbb{1} \tag{63}$$

In particular, the induced modality (61) *sends (direct) sums to (tensor) products*

$$!\big(\mathcal{H} \oplus \mathcal{H}'\big) \;\equiv\; \mathrm{QC}\big(\mathcal{H} \oplus \mathcal{H}'\big) \;\simeq\; \mathrm{Q}\big((\mathrm{C}\mathcal{H}) \times (\mathrm{C}\mathcal{H}')\big) \;\simeq\; (\mathrm{QC}\mathcal{H}) \times (\mathrm{QC}\mathcal{H}') \;\equiv\; (!\mathcal{H}) \otimes (!\mathcal{H}')$$

and zero (objects) to unit (objects)

$$!\,0 \;\equiv\; \mathrm{QC}\,0 \;\simeq\; \mathrm{Q}\ast \;\simeq\; \mathbb{1}$$

as befits an exponential map.

Proof. The adjunction itself is the composite of (60) with (57), as shown.

That Q is strong monoidal follows for instance from the fact that $\mathcal{H} \otimes (\text{-})$ is a left adjoint and hence distributes over the coproduct $\bigoplus_W$:

$$(\mathrm{Q}W) \otimes (\mathrm{Q}W') \;\equiv\; (\textstyle\bigoplus_W \mathbb{1}) \otimes (\textstyle\bigoplus_{W'} \mathbb{1}) \;\equiv\; \underset{W \times W'}{\bigoplus} (\mathbb{1} \otimes \mathbb{1}) \;=\; \underset{W \times W'}{\bigoplus} \mathbb{1} \;\equiv\; \mathrm{Q}(W \times W')\,.$$

Similarly, C is strong monoidal with respect to the Cartesian product on both sides, since $\natural(\mathbb{1} \to (\text{-}))$ is a right adjoint, whence it becomes lax monoidal with respect to the tensor product by composition with the universal bilinear map (55):

$$
\begin{array}{rcll}
(\mathrm{C}\mathcal{H}) \times (\mathrm{C}\mathcal{H}) & \equiv & \natural(\mathbb{1} \to \mathcal{H}) \times \natural(\mathbb{1} \to \mathcal{H}') & \\
& \simeq & \natural\Big((\mathbb{1} \to \mathcal{H}) \times (\mathbb{1} \to \mathcal{H}') \Big) & \text{since } \natural \text{ is right adjoint} \\
& \simeq & \natural\Big(\big(\mathbb{1} \to (\mathcal{H} \times \mathcal{H}')\big) \Big) & \text{since } \mathbb{1} \to (\text{-}) \text{ is right adjoint} \\
& \equiv & \mathrm{C}(\mathcal{H} \times \mathcal{H}') & \\
& \to & \mathrm{C}(\mathcal{H} \otimes \mathcal{H}') & \text{using (55)}
\end{array}
\qquad \square
$$

Remark 3.9 (Exponential modality). Prop 3.8 recovers — via dependent linear type formations — the *exponential modality* (16) usually postulated in linear logic/type theory (Lit. 2.4). In the model QuType $\equiv$ $\mathrm{Mod}_{\mathbb{C}}$ (50), the operation $\mathcal{H} \mapsto \natural(\mathbb{1} \to \mathcal{H})$ (60) produces the *underlying set* of vectors in the vector space $\mathcal{H}$, whence the exponential modality (61) sends a vector space to the linear span of its underlying set of vectors

$$
\mathcal{H} : \mathrm{Mod}_{\mathbb{C}} \qquad \vdash \qquad !\mathcal{H} = \bigoplus_{\mathcal{H}} \mathbb{1}
$$

Definition 3.10 (Quantization modality). We will regard quantization (61) as the *relative* monad obtained ([ACU15, Prop. 2.3]) by restricting the quantum-modality $\triangleright$ (3.6) along precomposition with (60):

$$
\begin{array}{cccccc}
\mathrm{Q} \;:\; & \mathrm{ClaType} & \xrightarrow{\mathbb{1}\times(\text{-})} & \mathrm{Type} & \xrightarrow{\triangleright} & \mathrm{Type} \\
& W & \mapsto & \begin{bmatrix} \mathbb{1}_\bullet \\ \downarrow \\ W \end{bmatrix} & \mapsto & \bigoplus_W \mathbb{1}
\end{array}
\tag{64}
$$

This (just) means that we take the `return`- and `bind`-operations (21) of Q to be special instances of those of $\triangleright$, as follows, where we use the linear type declaration from (56):

$$
\begin{array}{lll}
\mathtt{return}^{\mathrm{Q}}_{W} \;⦂\; \big(\mathbb{1} \times W \multimap \mathrm{Q}W\big) & \qquad & \mathtt{bind}^{\mathrm{Q}} \;⦂\; \big(\mathbb{1} \times W \multimap \mathrm{Q}W'\big) \multimap \big(\mathrm{Q}W \multimap \mathrm{Q}W'\big) \\
\mathtt{return}^{\mathrm{Q}}_{W} \;\equiv\; \mathtt{return}^{\triangleright}_{\mathbb{1}\times W} & \qquad & \mathtt{bind}^{\mathrm{Q}} \;\equiv\; \mathtt{bind}^{\triangleright}
\end{array}
$$

But in these special cases of $\triangleright$-operations we may, by (54), equivalently write this pleasantly suggestively as follows:

$$
\boxed{
\begin{array}{c||lll}
\textbf{quantized}\;\mathrm{Q} & \mathtt{return}^{\mathrm{Q}}_{W} \;⦂\; \big(W \to \mathrm{Q}W\big) & & \mathtt{bind}^{\mathrm{Q}} \;⦂\; \big(W \to \mathrm{Q}W'\big) \multimap \big(\mathrm{Q}W \longrightarrow\!\!\!\circ\; \mathrm{Q}W'\big) \\
& \mathtt{return}^{\mathrm{Q}}_{W} \;\equiv\; w \mapsto |w\rangle & & \mathtt{bind}^{\mathrm{Q}} \;\equiv\; \big(w \mapsto |\psi_w\rangle\big) \mapsto \Big(\sum\limits_{w} q_w |w\rangle \;\mapsto\; \sum\limits_{w} q_w |\psi_w\rangle\Big)
\end{array}
}
\tag{65}
$$

Hence the quantization monad, when handed a classical state w, `return`*s* the corresponding quantum state $|w\rangle$. In quantum information theory this is commonly used in the following:

Example 3.11 (Type of qbits). The notation for the quantization-monad (Def. 3.10) is such as to reproduce the standard notation "QBit" for the type of q-bits (eg. [NC10, §1.2], often also "qubit", eg. [Ri21]) as the quantum analog of the type Bit $\equiv \{0,1\}$ of classical bits (cf. [TQP, (110)]):

$$
\mathrm{QBit} \;\equiv\; \mathrm{Q}(\mathrm{Bit}) \;\equiv\; \triangleright(\mathbb{1}_{\mathrm{Bit}}) \;\equiv\; \bigoplus_{\mathrm{Bit}} \underset{\mathrm{Bit}}{\mathbb{1}} \;\equiv\; \bigoplus_{\{0,1\}} \underset{\{0,1\}}{\mathbb{1}} \;\equiv\; \mathbb{1}_{0} \oplus \mathbb{1}_{1} \;=\; \Big\{ q_0\,|0\rangle \;+\; q_1\,|1\rangle \Big\}.
\tag{66}
$$

The Quantum/Classical Divide

Modality	Idempotent **monad**	**Pure effect**
Classical	$\natural \;:\; \mathrm{Type} \twoheadrightarrow \mathrm{ClaType} \hookrightarrow \mathrm{Type}$ $\natural \equiv \left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \mapsto W \mapsto \left[\begin{array}{c} 0_\bullet \\ \downarrow \\ W \end{array}\right]$ (strong wrt ×)	$\mathrm{ret}^{\natural}_{\mathcal{H}_\bullet} \;:\; \mathcal{H}_\bullet \longrightarrow \natural\mathcal{H}_\bullet$ $\left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \begin{array}{c} \xrightarrow{0} \\ \xrightarrow{\mathrm{id}} \end{array} \left[\begin{array}{c} 0_\bullet \\ \downarrow \\ W \end{array}\right]$
Quantum	$\triangleright \;:\; \mathrm{Type} \twoheadrightarrow \mathrm{QuType} \hookrightarrow \mathrm{Type}$ $\triangleright \equiv \left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \mapsto \bigoplus_W \mathcal{H}_\bullet \mapsto \left[\begin{array}{c} \bigoplus_W \mathcal{H}_\bullet \\ \downarrow \\ * \end{array}\right]$ (strong wrt ⊗)	$\mathrm{ret}^{\triangleright}_{\mathcal{H}_\bullet} \mathrel{\substack{\circ \\ \circ}} \mathcal{H}_\bullet \multimap \triangleright\mathcal{H}_\bullet$ $\left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \begin{array}{c} \xrightarrow{\mathrm{ret}^{\lozenge_W}_{\mathcal{H}_\bullet}} \\ \xrightarrow{p_B} \end{array} \left[\begin{array}{c} \bigoplus_W \mathcal{H}_\bullet \\ \downarrow \\ * \end{array}\right]$
Quantized	$\mathrm{Q} \;:\; \mathrm{ClaType} \to \mathrm{QuType} \hookrightarrow \mathrm{Type}$ $\mathrm{Q} \equiv W \mapsto \triangleright(\mathbb{1}_W)$ (relative monad)	$\mathrm{ret}^{\mathrm{Q}}_{\mathcal{H}_\bullet} \mathrel{\substack{\circ \\ \circ}} W \longrightarrow \mathrm{Q}W$ $\left[\begin{array}{c} \mathbb{1}_\bullet \\ \downarrow \\ W \end{array}\right] \begin{array}{c} \xrightarrow{\mathrm{ret}^{\lozenge_B}_{\mathcal{E}}} \\ \xrightarrow{p_B} \end{array} \left[\begin{array}{c} \bigoplus_W \mathbb{1} \\ \downarrow \\ * \end{array}\right]$

Base change and dependent classical/linear type formation. In parameterized generalization of the reflection of quantum types inside all bundle types (Prop. 3.6), also the W-parameterized linear types (50) are reflective in the *slice category* $\mathrm{Type}_{/W}$ of bundle types over the given classical type $W = \left[\begin{array}{c} 0_\bullet \\ \downarrow \\ W \end{array}\right]$:

$$\begin{tikzcd}
\left[\begin{array}{c} \bigoplus\limits_{p'(w')=w} \mathcal{H}'_{w'} \\ \downarrow \\ (w : W) \end{array}\right] & \left[\begin{array}{ccc} \mathcal{H}'_\bullet & \to & W' \\ \downarrow & & \downarrow {\scriptstyle p'} \\ 0_\bullet & \to & W \end{array}\right] \arrow[l, mapsto] \\
\mathrm{QuType}_W \arrow[r, hook, shift right] \arrow[r, phantom, "\perp"] & \mathrm{Type}_{/W} \arrow[l, shift right] \arrow[loop right, "\triangleright_W"] \arrow[loop right, phantom, "\textcolor{orange}{\textbf{W-quantumly}}"', near end] \\
\left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \arrow[r, mapsto] & \left[\begin{array}{ccc} \mathcal{H}_\bullet & \to & W \\ \downarrow & & \downarrow \\ 0_\bullet & \to & W \end{array}\right]
\end{tikzcd} \tag{67}$$

But the category of linear bundle types is locally cartesian closed, in particular:

Proposition 3.12. *For* $W, \Gamma : \mathrm{ClaType}$ *and* $p : W \to \Gamma$*, the pullback base change operation* $W \times_\Gamma (\text{-})$ *between the respective slices of the category of linear bundle types (Def. 3.1)*

$$\begin{tikzcd}
W \arrow[rr, "p"] & & \Gamma \\
\mathrm{Type}_{/W} & & \mathrm{Type}_{/\Gamma} \arrow[ll, "W \times_\Gamma (\text{-})", "\textbf{contex extension}"'] \\
\left[\begin{array}{ccc} \mathcal{H}'_{w'} & \to & (w' : W'_{p(w)}) \\ \downarrow & & \downarrow \\ 0_w & \to & (w : W) \end{array}\right] & & \left[\begin{array}{ccc} \mathcal{H}'_\bullet & \to & W' \\ \downarrow & & \downarrow \\ 0_\bullet & \to & \Gamma \end{array}\right] \arrow[ll, mapsto]
\end{tikzcd}$$

has both a left adjoint ("dependent coproduct[17]") and a right adjoint ("dependent product"), given as follows:

$$\begin{tikzcd}
\left[\begin{array}{ccc} \mathcal{H}'_\bullet & \to & W' \\ \downarrow & & \downarrow {\scriptstyle p'} \\ 0_\bullet & \to & W \end{array}\right] \arrow[r, mapsto] & \left[\begin{array}{ccc} \mathcal{H}'_{w'_w} & \to & \Big((w, w'_w) : \coprod\limits_{p(w)=\gamma} W'_w\Big) \\ \downarrow & & \downarrow \\ 0_\bullet & \longrightarrow & (\gamma : \Gamma) \end{array}\right] \\
\mathrm{Type}_{/W} \arrow[r, shift left=3, "\textbf{dependent coproduct}", "\coprod_W"'] \arrow[r, phantom, shift left=1.5, "\perp"] \arrow[r, shift right=3, "\prod_W", "\textbf{dependent product}"'] \arrow[r, phantom, shift right=1.5, "\perp"] & \mathrm{Type}_{/\Gamma} \arrow[l, "W \times_\Gamma (\text{-})"'] \\
\left[\begin{array}{ccc} \mathcal{H}'_\bullet & \to & W' \\ \downarrow & & \downarrow \\ 0_\bullet & \to & W \end{array}\right] \arrow[r, mapsto] & \left[\begin{array}{ccc} \prod\limits_{p(w)=\gamma} \mathcal{H}'_{w'_w} & \to & \Big(w'_\bullet : \prod\limits_{p(w)=\gamma} W'_w\Big) \\ \downarrow & & \downarrow \\ 0_\bullet & \longrightarrow & (\gamma : \Gamma) \end{array}\right]
\end{tikzcd} \tag{68}$$

Proof. We may formally check the hom-isomorphisms, using (52). It is sufficient to consider the case that $\Gamma = *$:

$$\begin{aligned}
& \mathrm{Hom}\left(\left[\begin{array}{c} \mathcal{H}'_{w'_w} \\ \downarrow \\ \Big((w, w'_w) : \coprod_w W'_w\Big) \end{array}\right], \left[\begin{array}{c} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{array}\right] \right) \\
\simeq\; & \Big(f_\bullet : \coprod_w W'_w \to W''\Big) \times \prod_{(w, w'_w)} \natural\Big(\mathcal{H}_{w'_w} \to \mathcal{H}''_{f_w(w'_w)}\Big) \\
\simeq\; & \prod_w \Big(\big(f_w : W'_w \to W''\big) \times \prod_{w'_w} \natural\big(\mathcal{H}_{w'_w} \to \mathcal{H}''_{f_w(w'_w)}\big) \Big) \\
\simeq\; & \mathrm{Hom}_{/W}\left(\left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right], W \times \left[\begin{array}{c} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{array}\right] \right)
\end{aligned}
\qquad
\begin{aligned}
& \mathrm{Hom}\left(\left[\begin{array}{c} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{array}\right], \left[\begin{array}{c} \prod_w \mathcal{H}'_{w'_w} \\ \downarrow \\ w'_\bullet : \prod_w W'_w \end{array}\right] \right) \\
\simeq\; & \Big(f'_\bullet : W'' \to \prod_w W'_w\Big) \times \prod_{w''} \natural\Big(\mathcal{H}''_{w''} \to \prod_w \mathcal{H}'_{f'_w(w'')}\Big) \\
\simeq\; & \prod_w \Big(\big(f'_w : W'' \to W'_w\big) \times \prod_{w''} \natural\big(\mathcal{H}''_{w''} \to \mathcal{H}'_{f'_w(w'')}\big) \Big) \\
\simeq\; & \mathrm{Hom}_{/W}\left(W \times \left[\begin{array}{c} \mathcal{H}'' \\ \downarrow \\ W'' \end{array}\right], \left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right] \right)
\end{aligned}$$

□

---

[17] Of course, in type theory this dependent coproduct $\coprod_W$ is traditionally called the "dependent sum" and denoted "Σ_W". But this (quite unnecessary but deeply engrained) abuse of terminology/notation from linear algebra becomes problematic in the context of dependent linear type theory with its actual (direct) *sums* $\bigoplus_W$ of linear types.

The (co)restriction of the base change adjoint triple (68) along the reflective inclusion of W-quantum types (67) yields base change of dependent linear types:

$$\begin{tikzcd}[column sep=large]
\mathrm{Type}_{/W}
  \arrow[rr, shift left=16pt, leftarrow, "\coprod_W" description]
  \arrow[rr, phantom, shift left=8pt, "\perp"]
  \arrow[rr, leftarrow, "W \times_\Gamma (\text{-})" description]
  \arrow[rr, phantom, shift right=8pt, "\perp"]
  \arrow[rr, shift right=16pt, leftarrow, "\prod_W" description]
& & \mathrm{Type}_{/\Gamma}
  \arrow[dr, shift left=6pt, "\triangleright_\Gamma"]
  \arrow[dr, phantom, "\perp"]
& \\
& \mathrm{QuType}_W
  \arrow[ul, hook]
  \arrow[rr, shift left=16pt, leftarrow, "\oplus_W" description]
  \arrow[rr, phantom, shift left=8pt, "\perp"]
  \arrow[rr, leftarrow, "\underset{W}{\mathbb{1}} \otimes (\text{-})" description]
  \arrow[rr, phantom, shift right=8pt, "\perp"]
  \arrow[rr, shift right=16pt, leftarrow, "\prod_W" description]
& & \mathrm{QuType}_\Gamma
  \arrow[ul, hook, shift left=6pt]
\\
& (w : W \;\vdash\; \mathcal{H}_w) & \mapsto & \Big(\gamma : \Gamma \;\vdash\; \prod_{p(w)=\gamma} \mathcal{H}_w\Big)
\end{tikzcd} \tag{69}$$

Now something special happens: If W is *finite* (over Γ) then the direct sum and the direct product of linear spaces coincide, $\oplus_W \simeq \prod_W$, and so this adjunction on linear types becomes ambidextrous:

$$\Gamma : \mathrm{ClaType}, \quad W : \mathrm{FinClaType} \qquad \vdash \qquad \begin{tikzcd}[column sep=large]
\big(w : W \;\vdash\; \mathcal{H}_w\big) & \mapsto & \Big(\gamma : \Gamma \;\vdash\; \bigoplus_{p(w)=\gamma} \mathcal{H}_w\Big) \\
\mathrm{QuType}_W
  \arrow[rr, shift left=16pt, leftarrow, "\oplus_W" description]
  \arrow[rr, phantom, shift left=8pt, "\perp"]
  \arrow[rr, leftarrow, "\underset{W}{\mathbb{1}} \otimes (\text{-})" description]
  \arrow[rr, phantom, shift right=8pt, "\perp"]
  \arrow[rr, shift right=16pt, leftarrow, "\oplus_W" description]
& & \mathrm{QuType}_\Gamma
\end{tikzcd} \tag{70}$$

All these structures and properties are elementary to see in the concrete model of indexed sets of vector spaces, but they hold quite generally for (higher) categories of parameterized linear (homotopy) types. In fact, much of this structure is that traditionally encoded by *Grothendieck's yoga of six operations* used in motivic (homotopy) theory.

Motivic yoga. For the purposes of the present discussion we make the following definition (cf. [EoS, pp. 41]):

Definition 3.13 (Motivic Yoga). Let Type be a locally cartesian closed category with coproducts. We say that a *Grothendieck-Wirthmüller motivic yoga of operations* on Type – or just *motivic yoga*, for short – is:

(i) an ambidextrously reflected subcategory ClaType ("of classical base types"), hence a functor $\natural$ onto a full subcategory such that it is both left and right adjoint to the inclusion functor:

$$\begin{tikzcd} \mathrm{ClaType} \arrow[r, hook] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, phantom, shift right=2.5, "\perp"'] & \mathrm{Type} \arrow[l, shift right=5, "\natural"'] \arrow[l, shift left=5, "\natural"] \arrow[loop right, "\natural"] \end{tikzcd} \tag{71}$$

This implies in particular that ClaType has all (fiber-)products and coproducts, and we write

$$\mathrm{FinClaType} \hookrightarrow \mathrm{ClaType} \tag{72}$$

for the further full subcategory on the finite coproducts of the terminal object with itself.

(ii) For each $W\colon \mathrm{ClaType}$ a symmetric closed monoidal structure $(\mathrm{QuType}_B, \otimes_B, \mathbb{1}_B)$ on the iso-comma categories ("of linear bundles over W"):

$$\mathrm{QuType}_W \;\equiv\; \natural/W \;=\; \left\{ \begin{bmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{bmatrix} \begin{matrix} \xrightarrow{\;\phi_\bullet\;} \\ \\ = \!= \!= \end{matrix} \begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W \end{bmatrix} \right\}, \tag{73}$$

(iii) For each morphism in ClaType an adjoint triple of ("base change") functors:

$$\text{for} \quad B \xrightarrow{f} B' \quad \text{we have} \quad \begin{tikzcd} \mathrm{QuType}_W \arrow[r, shift left=5, "f_!"] \arrow[r, phantom, shift left=2.5, "\perp"] \arrow[r, shift right=5, "f_*"'] \arrow[r, phantom, shift right=2.5, "\perp"'] & \mathrm{QuType}_{W'} \arrow[l, "f^*"] \end{tikzcd} \tag{74}$$

such that the following conditions hold:

(a) **Linearity**: the left and right base change along finite types $W \xrightarrow{p_W} *$ (72) are naturally equivalent:

$$W \;:\; \mathrm{FinClaType} \qquad \vdash \qquad (p_w)_! \;\simeq\; (p_w)_*$$

(b) **Functoriality**: for composable morphisms f, g of base objects we have

$$(f^* \circ g^*) \;\simeq\; g^* \circ f^* \quad \text{and} \quad \mathrm{id}^* = \mathrm{id} \tag{75}$$

(c) **Monoidalness**: the pullback functors are strong monoidal in that there are natural equivalences:

$$f^*(\mathcal{H} \underset{W'}{\otimes} \mathcal{H}')_\bullet \;\simeq\; \big(f^*(\mathcal{H}) \underset{W'}{\otimes} f^*(\mathcal{H}')\big)_\bullet$$

(d) **Beck-Chevalley condition**: for a pullback square in ClaType the "pull-push operations" across one tip are naturally equivalent to those across the other:

$$\text{For} \quad \begin{tikzcd} & B \times_{B_0} B' \arrow[dl, "\mathrm{pr}_B"'] \arrow[dr, "\mathrm{pr}_{B'}"] & \\ B \arrow[dr, "p_B"'] & (\mathrm{pb}) & B' \arrow[dl, "p_{B'}"] \\ & B_0 & \end{tikzcd} \quad \text{we have} \quad \begin{tikzcd} & \mathrm{QuType}_{B \times_{B_0} B'} \arrow[dl, "(\mathrm{pr}_B)_!"'] & \\ \mathrm{QuType}_B & & \mathrm{QuType}_{B'} \arrow[ul, "(\mathrm{pr}_{B'})^*"'] \arrow[dl, "(p_{B'})_!"] \\ & \mathrm{QuType}_{B_0} \arrow[ul, "(p_B)^*"] & \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} & \mathrm{QuType}_{B \times_{B_0} B'} \arrow[dl, "(\mathrm{pr}_B)_*"'] & \\ \mathrm{QuType}_B & & \mathrm{QuType}_{B'} \arrow[ul, "(\mathrm{pr}_{B'})^*"'] \arrow[dl, "(p_{B'})_*"] \\ & \mathrm{QuType}_{B_0} \arrow[ul, "(p_B)^*"] & \end{tikzcd} \tag{76}$$

(e) **Frobenius reciprocity / projection formula**: the left pushforward of a tensor with a pullback is naturally equivalent to the tensor with the left pushforward:

$$f_!\big(\mathcal{H} \underset{W}{\otimes} f^*(\mathcal{H}')\big)_\bullet \;\simeq\; f_!(\mathcal{H}) \underset{W'}{\otimes} \mathcal{H}' \tag{77}$$

This is equivalent to f^* being also strong closed.

Proposition 3.14 (**Linear bundle types satisfy Motivic Yoga).** *The indexed category $W \mapsto \mathrm{QuType}_W$ of Def. 3.1 satisfies the motivic yoga (Def. 3.13) with respect to the fiberwise tensor product:*

$$
\begin{array}{ccc}
\mathrm{QuType}_W \times \mathrm{QuType}_W & \xrightarrow{\underset{W}{\otimes}} & \mathrm{QuType}_W \\
\left(\begin{bmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{bmatrix}, \begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W \end{bmatrix} \right) & \mapsto & \begin{bmatrix} \mathcal{H}_w \otimes \mathcal{H}'_w \\ \downarrow \\ (w : W) \end{bmatrix}
\end{array}
$$

Proof. This is straightforward to check. Details for this case and its higher generalization are also spelled out in [EoS]. ☐

We may alternatively see the monoidality of $\rhd$ just using the motivic yoga. For this purpose we shall denote the projection maps involves in a cartesian product as follows:

$$\begin{tikzcd}
 & W \times W' \arrow[dl, "\mathrm{pr}_W"'] \arrow[dr, "\mathrm{pr}_{W'}"] \arrow[dd, "p_{W \times W'}"] & \\
W \arrow[dr, "p_W"'] & & W' \arrow[dl, "p_{W'}"] \\
 & * &
\end{tikzcd} \tag{78}$$

Now:

$$
\begin{array}{rcll}
\rhd\big(\mathcal{E} \otimes \mathcal{E}'\big) & = & (p_{B \times B'})_! \big((\mathrm{pr}_B)^* \mathcal{E} \otimes (\mathrm{pr}_{B'})^* \mathcal{E}' \big) & \text{def} \\
 & \simeq & (p_B)_! (\mathrm{pr}_B)_! \big((\mathrm{pr}_B)^* \mathcal{E} \otimes (\mathrm{pr}_{B'})^* \mathcal{E}' \big) & (78) \\
 & \simeq & (p_B)_! \Big(\mathcal{E} \otimes \big((\mathrm{pr}_{B'})_! (\mathrm{pr}_{B'})^* E \big) \Big) & \text{Frob} \\
 & \simeq & (p_B)_! \Big(\mathcal{E} \otimes \big((p_B)^* (p_{B'})_! E \big) \Big) & \text{BC} \\
 & \simeq & \big((p_B)_! E \big) \otimes \big((p_{B'})_! \mathcal{E}' \big) & \text{Frob}\,.
\end{array}
$$

similarly:

$$
\begin{array}{rcll}
\mathrm{Q}(B \times B) & = & (p_{B \times B})_! (p_{B \times B})^* \mathbb{1} & \text{def} \\
 & \simeq & (p_{B'})_! (\mathrm{pr}_{B'})_! (\mathrm{pr}_B)^* (p_B)^* \mathbb{1} & (78) \\
 & \simeq & (p_{B'})_! (p_{B'})^* (p_B)_! (p_B)^* \mathbb{1} & \text{BC} \\
 & \simeq & (p_{B'})_! \big(\mathbb{1}_{B'} \otimes (p_{B'})^* (p_B)_! (p_B)^* \mathbb{1} \big) & \text{unit law} \\
 & \simeq & \big((p_{B'})_! \mathbb{1}_{B'} \big) \otimes \big((p_B)_! (p_B)^* \mathbb{1} \big) & \text{Frob} \\
 & \simeq & \big((p_{B'})_! (p_{B'})^* \mathbb{1} \big) \otimes \big((p_B)_! (p_B)^* \mathbb{1} \big) & \text{strong mon} \\
 & = & (\mathrm{Q}B) \otimes (\mathrm{Q}B') & \text{def}
\end{array}
$$

...The linear modality is idempotent $\rhd \rhd \xrightarrow[\sim]{\mu^{\rhd}} \rhd$.

(...)

(...edit and move or delete...)

Modal quantum logic of compound systems. With a linear data type thought of as representing the states of a given quantum system, we may think of the tensor product of two linear types as representing the states of the corresponding *compound quantum system*. The following properties of the tensor product, hence of compound quantum systems, are all *implied* by the simple axioms of dependent linear types (whence the "yoga of six functors").

(i) Frobenius reciprocity. For any $B \in \mathrm{Type}$, $\mathcal{H} \in \mathrm{LinType}_B$ and $\mathcal{K} \in \mathrm{Type}$, we have a natural equivalence of this form:

$$(p_B)_!\big(\mathcal{H}_\bullet \otimes (p_B)^*\mathcal{K}\big) \;\simeq\; \big((p_B)_!\mathcal{H}_\bullet\big) \otimes \mathcal{K} \tag{79}$$

(ii) Beck-Chevalley property. Given a pullback diagram of contexts

$$\begin{tikzcd}
 & B \underset{C}{\times} B' \arrow[dl, "\mathrm{pr}_B"'] \arrow[dr, "\mathrm{pr}_{B'}"] & \\
B \arrow[dr, "p_B"'] & & B' \arrow[dl, "p_{B'}"] \\
 & C &
\end{tikzcd}$$

we have a natural equivalence

$$(\mathrm{pr}_{B'})_! \circ (\mathrm{pr}_B)^* \;\simeq\; (p_B)^* \circ (p_B)_!\,. \tag{80}$$

We will consider this particularly in the case of plain products of contexts:

$$\begin{tikzcd}
 & B \times B' \arrow[dl, "\mathrm{pr}_B"'] \arrow[dr, "\mathrm{pr}_{B'}"] \arrow[dd, "p_{B\times B'}"] & \\
B \arrow[dr, "p_B"'] & & B' \arrow[dl, "p_{B'}"] \\
 & * &
\end{tikzcd}
\quad\vdash\quad
\begin{array}{rcl}
(\mathrm{pr}_B)^* \circ (\mathrm{p}_B)^* & \simeq & (\mathrm{pr}_{B'})^* \circ (\mathrm{p}_{B'})^* \\
(\mathrm{pr}_{B'})_! \circ (\mathrm{pr}_B)^* & \simeq & (p_{B'})^* \circ (p_B)_!
\end{array} \tag{81}$$

(iii) External tensor product. We set

$$\mathcal{H}_\bullet : \mathrm{LinType}_B\,,\;\; \mathcal{H}'_\bullet : \mathrm{LinType}_{B'} \;\;\vdash\;\; \overset{\substack{\text{external}\\ \text{tensor product}}}{\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet} \;:=\; \big((\mathrm{pr}_B)^*\mathcal{H}_\bullet\big) \otimes \big((\mathrm{pr}_{B'})^*\mathcal{H}'_\bullet\big) \tag{82}$$

$$\begin{tikzcd}
 & \mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet \arrow[d] & \\
 & B \times B' \arrow[ddl, "\mathrm{pr}_B"'] \arrow[ddr, "\mathrm{pr}_{B'}"] & \\
\mathcal{H}_\bullet \arrow[d] & & \mathcal{H}'_\bullet \arrow[d] \\
B & & B'
\end{tikzcd}$$

(a) The external tensor product (82) is respected by left base change, in that:

$$\boxed{(p_{B\times B'})_!\big(\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet\big) \;\simeq\; \big((p_B)_!\mathcal{H}_\bullet\big) \otimes \big((p_{B'})_!\mathcal{H}'_\bullet\big) \;=:\; \mathcal{H} \otimes \mathcal{H}'.} \tag{83}$$

Proof.

$$\begin{aligned}
(p_{B\times B'})_!\big(\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet\big) &\simeq (p_{B\times B'})_!\Big(\big((\mathrm{pr}_B)^*\mathcal{H}_\bullet\big) \otimes \big((\mathrm{pr}_{B'})^*\mathcal{H}'_\bullet\big)\Big) && \text{by (82)} \\
&\simeq (p_B)_!(\mathrm{pr}_B)_!\Big(\big((\mathrm{pr}_B)^*\mathcal{H}_\bullet\big) \otimes \big((\mathrm{pr}_{B'})^*\mathcal{H}'_\bullet\big)\Big) && \text{by (81)} \\
&\simeq (p_B)_!\Big(\mathcal{H}_\bullet \otimes \big((\mathrm{pr}_B)_!(\mathrm{pr}_{B'})^*\mathcal{H}'_\bullet\big)\Big) && \text{by (79)} \\
&\simeq (p_B)_!\Big(\mathcal{H}_\bullet \otimes \big((p_B)^*(p_{B'})_!\mathcal{H}'_\bullet\big)\Big) && \text{by (80)} \\
&\simeq \Big(\big((p_B)_!\mathcal{H}_\bullet\big) \otimes \big((p_{B'})_!\mathcal{H}'_\bullet\big)\Big) && \text{by (79)}
\end{aligned}$$

□

This induces

(b) The external tensor product (82) is respected by the possibility (and hence the necessity) modalities, in that:

$$\boxed{\Diamond_{B\times B'}\big(\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet\big) \;\simeq\; \big(\Diamond_B \mathcal{H}_\bullet\big) \boxtimes \big(\Diamond_{B'} \mathcal{H}_\bullet\big).}$$

Proof.

$$\begin{aligned}
\lozenge_{B\times B'}\big(\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet\big) &= (p_{B\times B'})^*(p_{B\times B'})_!\big(\mathcal{H}_\bullet \boxtimes \mathcal{H}'_\bullet\big) \\
&\simeq (p_{B\times B'})^*\Big(\big((p_B)_!\mathcal{H}_\bullet\big)\otimes\big((p_B)_!\mathcal{H}'_\bullet\big)\Big) && \text{by (83)} \\
&\simeq \big((p_{B\times B'})^*(p_B)_!\mathcal{H}_\bullet\big)\otimes\big((p_{B\times B'})^*(p_{B'})_!\mathcal{H}'_\bullet\big) && \text{by (\textbf{??})} \\
&\simeq \big((\mathrm{pr}_B)^*(p_B)^*(p_B)_!\mathcal{H}_\bullet\big)\otimes\big((\mathrm{pr}_{B'})^*(p_{B'})^*(p_{B'})_!\mathcal{H}'_\bullet\big) && \text{by (81)} \\
&\simeq \big((\mathrm{pr}_B)^*\square_B\mathcal{H}_\bullet\big)\otimes\big((\mathrm{pr}_{B'})^*\square_{B'}\mathcal{H}'_\bullet\big) && \text{by def} \\
&\simeq \big(\square_B\mathcal{H}_\bullet\big)\boxtimes\big(\square_{B'}\mathcal{H}'_\bullet\big) && \text{by (82) .}
\end{aligned}$$

□

3.2 Quantum syntax

We give an exposition of some of the formal syntax of `LHoTT` due to [RFL21][Ri22], matched to its denotational semantics in the 1-categories of linear bundle types from §3.1 and more generally in the simplicial categories of simplicial local systems discussed in [EoS]. While previous indication of the intended categorical semantics in [RFL21, §7.1] is still rather syntactical, we aim to unwind the actual diagrams which interpret given dependent type declarations in the target category.

This is to indicate by example how `LHoTT` is indeed a formal type theory for all the constructions considered in hupf, but an exhaustive treatment of this claim needs to be given elsewhere.

§3.2.1: Category theory of bireflective Frobenius monads
§3.2.2: Basic inference rules and their Categorical semantics
§3.2.3: Syntactic representation of the Motivic Yoga

Throughout, we make extensive use of the *pasting law*, which says that for a pasting diagram of two commuting squares in any category where the right square is cartesian, then two total rectangle is cartesian if and only if also the left square is cartesian:

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d] \arrow[rr, bend left, "\text{(pb)}"] & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \bullet \arrow[d] & & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \bullet \arrow[d] \\
\bullet \arrow[r] & \bullet \arrow[r] & \bullet \arrow[r, phantom, "\Leftrightarrow"] & & \bullet \arrow[r] & \bullet \arrow[r] & \bullet
\end{tikzcd} \tag{84}$$

3.2.1 Background: Bireflective Frobenius monads

The first layer of new type inference rules that `LHoTT` adjoins to plain `HoTT` is axioms for the classical-modality $\natural$ (57), hence the *infinitesimal cohesive modality* (Lit. 2.10). As a (co)monadic modality (Lit. 2.14) it is special in that it constitutes a *bireflective Frobenius monad* (59).

Therefore, in preparation of the semantic rules below in §3.2.2, we recall and develop some basic category theory of bireflective Frobenius monads. The reader may not want to go through this material linearly, we will point back to here where necessary.

Semantics of lex (ambidextrous) modalities. Write $\mathcal{T}$ for the interpreting (model) category.
Fact. A monad $\bigcirc : \mathcal{T} \to \mathcal{T}$, $\mathrm{ret}^{\bigcirc} : \mathrm{id} \to \bigcirc$ being *idempotent* with modal subcategory $\iota : \mathcal{T}^{\bigcirc} \hookrightarrow \mathcal{T}$ means that there are natural bijections

$$\begin{aligned} \mathrm{Hom}_{\mathcal{T}}\big(\bigcirc A,\, \iota(B)\big) &\xrightarrow{\;\;\sim\;\;} \mathrm{Hom}_{\mathcal{T}}\big(A,\, \iota(B)\big) \\ \big(\bigcirc A \xrightarrow{f} \iota(B)\big) \quad &\mapsto \quad \big(A \xrightarrow{\mathrm{ret}^{\bigcirc}_A} \bigcirc A \xrightarrow{f} \iota(B)\big) \end{aligned}$$

Dually, a comonad $\Box : \mathcal{T} \to \mathcal{T}$ being idempotent means that

$$\begin{aligned} \mathrm{Hom}_{\mathcal{T}}\big(\iota(B),\, \Box A\big) &\xrightarrow{\;\;\sim\;\;} \mathrm{Hom}_{\mathcal{T}}\big(\iota(B),\, A\big) \\ \big(\iota(B) \xrightarrow{f} \Box A\big) \quad &\mapsto \quad \big(\iota(B) \xrightarrow{f} \Box A \xrightarrow{\mathrm{obt}^{\Box}_A} A\big) \end{aligned}$$

Fact. If $\bigcirc : \mathcal{T} \to \mathcal{T}$ with unit $\mathrm{ret}^{\bigcirc} : \mathrm{id} \to \bigcirc$ is a lex modality on the ambient (model) category, then for each (fibrant) $\Gamma \in \mathcal{T}$ its induced lex modality on the (fibrational) slice $\mathcal{T}_{/\Gamma}$ is given by

$$\begin{array}{ccc} \bigcirc_\Gamma \;:\; & \mathcal{T}_{/\Gamma} \longrightarrow \mathcal{T}_{/\Gamma} \\ & \left[\begin{array}{c} A \\ \downarrow\!\!\!\downarrow p_A \\ \Gamma \end{array}\right] \;\mapsto\; \left[\begin{array}{c} (\mathrm{ret}^{\bigcirc}_A)^* A \\ \downarrow\!\!\!\downarrow \\ \Gamma \end{array}\right] & \text{where} \end{array} \qquad \begin{tikzcd} (\mathrm{ret}^{\bigcirc}_A)^* A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \bigcirc A \arrow[d, two heads, "\bigcirc p_A"] \\ \Gamma \arrow[r, "\mathrm{ret}^{\bigcirc}_{\Gamma}" description] & \bigcirc\Gamma \end{tikzcd} \tag{85}$$

with fiberwise unit $\mathrm{ret}^{\bigcirc_\Gamma}$ given by the canonical factorization of the global unit $\mathrm{ret}^{\bigcirc}_A$ through the defining pullback on the right:

$$\begin{tikzcd} A \arrow[r, dashed, "\mathrm{ret}^{\bigcirc_\Gamma}_A"] \arrow[rr, bend left, "\mathrm{ret}^{\bigcirc}_A" description] \arrow[d, two heads, "p_A"] & (\mathrm{ret}^{\bigcirc}_A)^* A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \bigcirc A \arrow[d, two heads, "p_A"] \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\bigcirc}_{\Gamma}" description] & \bigcirc\Gamma \end{tikzcd} \tag{86}$$

Proof. The technical ingredients underlying this statement all go back to [CHK85][CJKP97]; the statement as such is more explicit around [RSS20, Lem. 1.52, Thm. 1.54, Thm. A.9]. ☐

Remark 3.15 (Relative monads)**.** Instead of considering the full fiberwise monads $\natural_\Gamma : \mathcal{T}_{/\Gamma} \to \mathcal{T}_{/\Gamma}$, we want to restrict their formation to objects in $\mathcal{T}_{/\natural\Gamma}$, for reasons discussed in [RFL21, §1.2]. We now observe that this means to consider relative monads induced by $\natural_\Gamma$ (the actual monad will be recovered as $\natural\underline{(-)}$, see (118)).

Notation 3.16 (Full pullback along unit)**.** Given $p_A : A \to \natural\Gamma$, we denote its pullback along the $\natural$-unit of Γ by:

$$\begin{tikzcd} (\mathrm{ret}^{\natural}_A)^* A \arrow[r, "q_A"] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & A \arrow[d, "p_A"] \\ \natural\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}" description] & \Gamma \end{tikzcd} \tag{87}$$

Proposition 3.17. *For* $\Gamma \in \mathcal{T}$*, we obtain a* relative monad [ACU15, Def. 2.2] *with underlying functor*

$$\begin{tikzcd}
\natural^{\mathrm{rel}}_{\Gamma} \; : & \mathcal{T}_{/\natural\Gamma} \arrow[r, "(\mathrm{ret}^{\natural}_{\Gamma})^*"] & \mathcal{T}_{/\Gamma} \arrow[r, "\natural_{\Gamma}"] & \mathcal{T}_{/\Gamma}
\end{tikzcd}$$

$$\left[\begin{tikzcd} A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \end{tikzcd}\right] \mapsto \left[\begin{tikzcd} (\mathrm{ret}^{\natural}_{A})^* A \arrow[r, "q_A"] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & A \arrow[d, two heads, "p_A"] \\ \Gamma \arrow[r, dashed, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \end{tikzcd}\right] \mapsto \left[\begin{tikzcd} \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \natural(\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[r, "\natural q_A", "\sim"'] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural \mathrm{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}\right] \tag{88}$$

and with relative unit

$$\mathrm{ret}^{\natural^{\mathrm{rel}}_{\Gamma}}_{A} \;:=\; \mathrm{ret}^{\natural_{\Gamma}}_{(\mathrm{ret}^{\natural}_{\Gamma})^* A} \,. \tag{89}$$

Proof. This is an instance of [ACU15, Prop. 2.3 (1)]. ☐

Lemma 3.18 (Classical unit on pullback)**.** *The* $\natural$*-unit of* $(\mathrm{ret}^{\natural}_{\Gamma})^* A$ *in* (87) *equals the following composite:*

$$\begin{tikzcd}
(\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[r, "q_A"] \arrow[rrr, bend left, "\mathrm{ret}^{\natural}_{(\mathrm{ret}^{\natural}_{\Gamma})^* A}"] & A \arrow[r, "\mathrm{ret}^{\natural}_{A}"] & \natural A \arrow[r, "(\natural q_A)^{-1}"] & \natural\big((\mathrm{ret}^{\natural}_{\Gamma})^* A\big)\,,
\end{tikzcd} \tag{90}$$

where we use that $\natural q_A$ *is invertible, it being a pullback of* $\natural \mathrm{ret}^{\natural}_{\Gamma}$ *(since* $\natural$ *preserves pullbacks) which is invertible (since* $\natural$ *is idempotent).*

Proof. We may equivalently show that its $\beta \dashv \iota$ adjunct is the identity morphism. A priori, this adjunct equals the total top and right morphism in the following diagram:

$$\begin{tikzcd}
\beta\big((\mathrm{ret}^{\natural}_{\Gamma})^* A\big) \arrow[r, "\beta q_A"] & \beta A \arrow[r, "\beta \mathrm{ret}^{\iota\beta}_{A}"] \arrow[dr, equal] & \beta\iota\beta A \arrow[r, "\beta\iota(\beta q_A)^{-1}"] \arrow[d, "\mathrm{obt}^{\iota\beta}_{\beta A}"] & \beta\iota\beta\big((\mathrm{ret}^{\natural}_{A})^* A\big) \arrow[d, "\mathrm{obt}^{\iota\beta}_{\beta(\mathrm{ret}^{\natural}_{A})^* A}"] \\
 & & \beta A \arrow[r, "(\beta q_A)^{-1}"] & \beta\big((\mathrm{ret}^{\natural}_{A})^* A\big)
\end{tikzcd}$$

Here the square on the right commutes by naturality of the counit, and the triangle commutes by the triangle identity of the adjunction. Therefore the morphism in question equals the total bottom morphism, which is manifestly equal to the desired identity. ☐

Lemma 3.19 (Components of the relative monad)**.** *The relative unit of the* $(\mathrm{ret}^{\natural}_{\Gamma})^*$*-relative monad* (88) *has as components the unique dashed morphisms making the following diagrams commute:*

$$\begin{tikzcd}
(\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[rr, "q_A"] \arrow[d, dashed, "\mathrm{ret}^{\natural^{\mathrm{rel}}_{\Gamma}}_{A}"] \arrow[dr, "\mathrm{ret}^{\natural}_{(\mathrm{ret}^{\natural}_{\Gamma})^* A}"] \arrow[dd, bend right=60, "p_{(\mathrm{ret}^{\natural}_{\Gamma})^* A}"'] & & A \arrow[d, "\mathrm{ret}^{\natural}_{A}"] \\
\natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \natural(\mathrm{ret}^{\natural}_{A})^* A \arrow[r, "\natural q_A"] \arrow[d, "p_{(\mathrm{ret}^{\natural}_{\Gamma})^* A}"] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, "\natural p_A"] \\
\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural \mathrm{ret}^{\natural}_{A}"] & \natural\natural\Gamma
\end{tikzcd} \tag{91}$$

Proof. The point is that, by Lemma 3.18, the diagonal morphism is indeed a component of the $\natural$-unit as shown, making the top right square commute. With this the claim follows by (91) and (86). ☐

Bireflective Frobenius monads.

Definition 3.20. A *bireflective subcategory* inclusion in the sense of [FHPTST99, Def. 8] is an ambidextrously reflective subcategory inclusion

$$\begin{tikzcd}
\mathcal{C} \arrow[r, hook, "\iota" description] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"] & \mathcal{B} \arrow[l, shift right=4, "\beta"'] \arrow[l, shift left=4, "\beta"] \arrow[loop right, "\natural"]
\end{tikzcd} \qquad \text{such that :} \qquad \begin{tikzcd}
\natural \arrow[r, "\mathrm{obt}^{\natural}"] \arrow[dr, equal] & \mathrm{id}_{\mathcal{B}} \arrow[d, "\mathrm{ret}^{\natural}"] \\
 & \natural
\end{tikzcd} \tag{92}$$

Remark 3.21 (Idempotence)**.** Given a bireflective subcategory, the natural transformation

$$\mathrm{obt}^\natural \circ \mathrm{ret}^\natural \;:\; \mathrm{id}_{\mathcal{B}} \xrightarrow{\mathrm{ret}^\natural} \natural \xrightarrow{\mathrm{obt}^\natural} \mathrm{id}_\natural$$

is an idempotent endomorphism of the functor $\mathrm{id}_{\mathcal{B}}$. Together with the naturality of this transformation, it follows that for any morphism $\Gamma \xrightarrow{f} A$ in $\mathcal{B}$ its composites of the form $\mathrm{ret}^\natural_A \circ f$ are preserved by pre-composition with the idempotent, in that the following diagram commutes:

$$\begin{tikzcd}
\Gamma \arrow[r, "\mathrm{ret}^\natural_\Gamma"] \arrow[d, "f"'] & \natural\Gamma \arrow[r, "\mathrm{obt}^\natural_\Gamma"] & \Gamma \arrow[d, "f"] \\
A \arrow[r, "\mathrm{ret}^\natural_A"] & \natural A \arrow[r, "\mathrm{obt}^\natural_A"] \arrow[dr, equal] & A \arrow[d, "\mathrm{ret}^\natural_A"] \\
& & \natural A
\end{tikzcd} \tag{93}$$

Notation 3.22 (Pullback along counit)**.** For a bireflective subcategory and given $p_A \;:\; A \to \Gamma$, we write $\underline{A}$ for the pullback along the $\natural$-counit of Γ:

$$\begin{tikzcd}
\underline{A} \arrow[r, "v_A"] \arrow[d, "p_{\underline{A}}"'] \arrow[dr, phantom, "\text{(pb)}"] & A \arrow[d, "p_A"] \\
\natural\Gamma \arrow[r, "\mathrm{obt}^\natural_\Gamma"] & \Gamma
\end{tikzcd} \tag{94}$$

With the same kind of proof as for Lemma 3.18, we obtain:

Lemma 3.23 (Classical unit on ϵ-pullback)**.** *Given $p_A : A \to \Gamma$, then the $\natural$-unit on an object $\underline{A}$* (94) *equals the following composite:*

$$\begin{tikzcd}
\underline{A} \arrow[r, "v_A"] \arrow[rrr, bend left, "\mathrm{ret}^\natural_{\underline{A}}"] & A \arrow[r, "\mathrm{ret}^\natural_A"] & \natural A \arrow[r, "(\natural v_A)^{-1}"] & \natural\underline{A}\,.
\end{tikzcd} \tag{95}$$

Lemma 3.24. *Given $p_A : A \to \natural\Gamma$ we have $\underline{(\mathrm{ret}^\natural_A)^* A} \simeq A$.*

Proof. By the Pasting Law,

$$\begin{tikzcd}
A \arrow[drr, "p_A"'] & \simeq & \underline{(\mathrm{ret}^\natural_\Gamma)^* A} \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & (\mathrm{ret}^\natural_\Gamma)^* A \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & A \arrow[d, "p_A"] \\
& & \natural\Gamma \arrow[r, "\mathrm{obt}^\natural_\Gamma"] \arrow[rr, bend right, "\mathrm{id}"'] & \Gamma \arrow[r, "\mathrm{ret}^\natural_\Gamma"] & \natural\Gamma
\end{tikzcd} \tag{96}$$

□

Lemma 3.25 ([RFL21, Lem. 7.7])**.** *Given a bireflective subcategory inclusion (Def. 3.20), we have identifications*

$$\natural\big(\mathrm{ret}^\natural_{(-)}\big) \;=\; \mathrm{ret}^\natural_{\natural(-)} \qquad \textit{and} \qquad \natural\big(\mathrm{obt}^\natural_{(-)}\big) \;=\; \mathrm{obt}^\natural_{\natural(-)}\,. \tag{97}$$

Proof. Using the naturality squares of the unit over itself

$$\begin{tikzcd}
E \arrow[r, "\eta^\natural_E"] \arrow[d, "\eta^\natural_E"'] & \natural E \arrow[d, "\natural\eta^\natural_E"] \\
\natural E \arrow[r, "\eta^\natural_{\natural E}"] & \natural\natural E
\end{tikzcd} \tag{98}$$

we have

$$\natural\big(\mathrm{ret}^\natural_E\big) \underset{(92)}{=} \natural\big(\mathrm{ret}^\natural_E\big) \circ \mathrm{ret}^\natural_E \circ \mathrm{obt}^\natural_E \underset{(98)}{=} \mathrm{ret}^\natural_{\natural E} \circ \mathrm{ret}^\natural_E \circ \mathrm{obt}^\natural_E \underset{(92)}{=} \mathrm{ret}^\natural_{\natural E}\,.$$

An analogous argument proves the other case. □

Lemma 3.26 (Relations)**.** *Given a bireflective subcategory inclusion (Def. 3.20), we have*

$$\begin{tikzcd}
\natural\Gamma \arrow[r, "\natural\mathrm{ret}^\natural_\Gamma"] \arrow[rr, bend right, "\mathrm{id}"'] & \natural\natural\Gamma \arrow[r, "\mathrm{obt}^\natural_{\natural\Gamma}"] & \natural\Gamma
\end{tikzcd} \qquad \textit{hence, by (97), also:} \qquad \begin{tikzcd}
\natural\Gamma \arrow[r, "\mathrm{ret}^\natural_{\natural\Gamma}"] \arrow[rr, bend right, "\mathrm{id}"'] & \natural\natural\Gamma \arrow[r, "\mathrm{obt}^\natural_{\natural\Gamma}"] & \natural\Gamma
\end{tikzcd} \tag{99}$$

and so, since $\mathrm{obt}^\natural_{\natural\Gamma} = \natural\mathrm{obt}^\natural_\Gamma$ is an isomorphism by idempotency of $\natural$:

$$\mathrm{ret}^\natural_{\natural\Gamma} \;=\; \big(\mathrm{obt}^\natural_{\natural\Gamma}\big)^{-1}\,. \tag{100}$$

Proof. The following square commutes by the naturality of the counit

$$\begin{tikzcd}
\natural\Gamma \arrow[r, "\natural\mathrm{ret}^{\natural}_{\Gamma}"] \arrow[d, "\mathrm{obt}^{\natural}_{\Gamma}"'] \arrow[dr, equal] & \natural\natural\Gamma \arrow[d, "\mathrm{obt}^{\natural}_{\natural\Gamma}"] \\
\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"'] & \natural\Gamma
\end{tikzcd}$$

and the bottom left triangle commutes by (92). Therefore the top right triangle commutes. ☐

So in generalization of (93), we have:

Corollary 3.27 (Precomposition with projection)**.** *Given a bireflective subcategory inclusion (Def. 3.20), we have for* $f : \Gamma \to \natural A$ *that precomposition with* $\mathrm{obt}^{\natural}_{\Gamma} \circ \mathrm{ret}^{\natural}_{\Gamma}$ *acts like the identity:*

$$\begin{tikzcd}
\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] \arrow[d, "f"'] & \natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\Gamma}"] & \Gamma \arrow[d, "f"] \\
\natural A \arrow[r, "\mathrm{ret}^{\natural}_{\natural A}"] \arrow[rr, bend right, "\mathrm{id}"', "(99)"] & \natural\natural A \arrow[r, "\mathrm{obt}^{\natural}_{\natural A}"] & \natural A
\end{tikzcd} \tag{101}$$

We list the inference rules of Linear Homotopy Type Theory (`LHoTT`) together with their intended 1-categorical semantics (intended to be thought of as categories of linear bundles).

Previously [RFL21, §7] have indicated intended semantics (of the fragment excluding the tensor products) in "categories with families", in a form that still quite syntactic (linear strings of symbols). Here we show the actual diagrams in the interpreting category which lend themselves to usual category-theoretic arguments — cf. for instance our proof of the $\natural$-computation rules in (115) (116) with the corresponding argument in [RFL21, Lem. 7.11 (4) (5)]).

3.2.2 Basic `LHoTT` Inference rules and their categorical semantics

We showcase the most basic inference rules of `LHoTT` [RFL21][Ri22] and give their categorical semantics.

Dependent terms of dependent types. For reference and to introduce our notation, first to recall some standard inference rules of dependent types, cast in the following fashion:

Syntax	Semantics
$\gamma : \Gamma \;\vdash\; A_\gamma : \mathrm{Type}$ dependent type	$$\begin{tikzcd} (\gamma{:}\Gamma)\times A_\gamma \;\equiv\; A \arrow[r] \arrow[d, two heads, "p_A", "\text{display map}"'] \arrow[dr, phantom, "\text{(pb)}"] & \widehat{\mathrm{Obj}} \arrow[d, two heads] \\ \Gamma \arrow[r, dashed, "\vdash A", "\text{name of } A"'] & \mathrm{Obj} \;\; \text{object classifier} \end{tikzcd}$$
$\gamma : \Gamma \;\vdash\; a_\gamma : A_\gamma$ dependent term	$$\begin{tikzcd} \Gamma \arrow[r, dashed, "\vdash a" description, "\text{name of } a"] \arrow[d, equal] & A \arrow[d, two heads, "p_A"] \\ \Gamma \arrow[r, equal] & \Gamma \;\; \text{context} \end{tikzcd}$$

In analogous fashion we now have the following inference rules for dependent ♮-types:
Structural rules for general variables. (14)

Syntax	Semantics
VAR $\dfrac{\gamma : \Gamma \;\vdash\; A_\gamma : \mathrm{Type}}{\gamma : \Gamma,\;\; a_\gamma : A_\gamma,\;\; \Gamma' \;\vdash\; a_\gamma : A_\gamma}$ variable rule	$$\begin{tikzcd} \Gamma' \arrow[rrr, "p_{\Gamma'}"] \arrow[dr, dashed, "\vdash(-)", "\text{generic element}"'] \arrow[dd, equal] & & & A \arrow[dd, two heads, "p_A"] \\ & A_{\Gamma'} \arrow[urr] \arrow[d, two heads] \arrow[drr, phantom, "\text{(pb)}"] & & \\ \Gamma' \arrow[r, equal] & \Gamma' \arrow[r, "p_{\Gamma'}"] & A \arrow[r, "p_A"] & \Gamma \end{tikzcd}$$ (102)
W $\dfrac{\Gamma,\; \Delta \;\vdash\; j : J \qquad \Gamma \;\vdash\; A : \mathrm{Type}}{\Gamma,\; a{:}A,\; \delta{:}\Delta \;\vdash\; j_\delta : J_\delta}$ weakening rule	$$\begin{tikzcd} A \times_\Gamma \Delta \arrow[rrr] \arrow[drr, dashed, "\vdash j", "\text{pullback along display}"'] \arrow[dd, equal] & & & \Delta \arrow[r, "\vdash j"] \arrow[dd, equal] & J \arrow[dd, two heads, "p_J"] \\ & & p_A^* J \arrow[urr] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & \\ A \times_\Gamma \Delta \arrow[rr, equal] & & A \times_\Gamma \Delta \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \Delta \arrow[r, equal] \arrow[d, two heads, "p_\Delta"] & \Delta \\ & & A \arrow[r, "p_A"] & \Gamma & \end{tikzcd}$$ (103)
S $\dfrac{\begin{array}{r} \gamma{:}\Gamma,\; a_\gamma{:}A_\gamma,\; \delta_{a_\gamma}{:}\Delta_{a_\gamma} \;\vdash\; j_{\delta_{a_\gamma}} : J_{\delta_{a_\gamma}} \\ \gamma{:}\Gamma \;\vdash\; a^0_\gamma : A_\gamma \end{array}}{\gamma{:}\Gamma,\; \delta_{a^0_\gamma}{:}\Delta_{a^0_\gamma} \;\vdash\; j_{\delta_{a^0_\gamma}} : J_{\delta_{a^0_\gamma}}}$ substitution rule	$$\begin{tikzcd} \Delta_{a^0} \arrow[rrr] \arrow[drr, dashed, "\vdash j[a^0/a]", "\text{pullback along name}"'] \arrow[dd, equal] & & & \Delta \arrow[r, "\vdash j"] \arrow[dd, equal] & J \arrow[dd, two heads, "p_J"] \\ & & J_{[a^0/a]} \arrow[urr] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & \\ \Delta_{a^0} \arrow[rr, equal] & & \Delta_{a^0} \arrow[r, "{[a^0/a]}"] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \Delta \arrow[r, equal] \arrow[d, two heads, "p_\Delta"] & \Delta \\ & & \Gamma \arrow[r, "\vdash a^0"] \arrow[d, equal] & A \arrow[d, two heads, "p_A"] & \\ & & \Gamma \arrow[r, equal] & \Gamma & \end{tikzcd}$$ (104)

Structural rules for $\natural$-variables. (Syntax from [RFL21, Fig. 1][Ri22, Fig. 1.1])

Syntax	Semantics	
$\natural\text{-CTX}\ \dfrac{\Gamma\ \ \text{ctx}}{\underline{\Gamma}\ \ \text{ctx}}$ $\natural$-context rule	$$\frac{\Gamma}{\natural\Gamma}$$	(105)
$\natural\text{-CTX-EXT}\ \dfrac{\Gamma\ \ \text{ctx} \qquad \underline{\Gamma}\ \vdash\ A : \text{Type}}{\Gamma,\ \underline{a}{:}A\ \ \text{ctx}}$ relative $\natural$-context rule	$$\frac{\begin{tikzcd} A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \end{tikzcd}}{\begin{tikzcd} \text{(88)}\ \natural^{\text{rel}}_{\Gamma} A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural(\text{ret}^{\natural}_{\Gamma})^{*} A \arrow[r, "\natural q_A"] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ \Gamma \arrow[r, "\text{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\text{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}}$$	(106)
$\dfrac{\Gamma\ \vdash\ a : A}{\underline{\Gamma}\ \vdash\ \underline{a} : \underline{A}}$	$$\begin{tikzcd} \natural\Gamma \arrow[rr, "\text{obt}^{\natural}_{\Gamma}"] \arrow[dr, dashed, "\vdash\underline{a}"] \arrow[dd, equal] & & \Gamma \arrow[r, "\vdash a"] \arrow[dd, equal] & A \arrow[dd, two heads, "p_A"] \\ & \underline{A} \arrow[urr] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & \\ \natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\text{obt}^{\natural}_{\Gamma}"] & \Gamma \arrow[r, equal] & \Gamma \end{tikzcd}$$	(107)
$\dfrac{\underline{\Gamma}\ \vdash\ a : A}{\underline{\Gamma}\ \vdash\ \underline{a} \equiv a : A}$	$$\begin{tikzcd} \natural\Gamma \arrow[rrr, bend left, "\text{id}", "\text{(92)}"'] \arrow[rr, "\text{obt}^{\natural}_{\Gamma}"] \arrow[dr, dashed, "\vdash\underline{a}"] \arrow[dd, equal] & & \Gamma \arrow[r, "\text{ret}^{\natural}_{\Gamma}"] \arrow[dd, equal] & \natural\Gamma \arrow[r, "\vdash a"] \arrow[dd, equal] & A \arrow[dd, two heads, "p_A"] \\ & A \arrow[urrr, equal] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & & \\ \natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\text{obt}^{\natural}_{\Gamma}"] \arrow[rr, bend right, "\text{id}"', "\text{(92)}"] & \Gamma \arrow[r, "\text{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, equal] & \natural\Gamma \end{tikzcd}$$	(108)
$\natural\text{VAR}\ \dfrac{\underline{\gamma} : \underline{\Gamma}\ \vdash\ A_{\underline{\gamma}}{:}\text{Type}}{\gamma{:}\Gamma,\ \ \underline{a}_{\gamma}{:}A_{\gamma},\ \ \Gamma'_{\underline{a}_{\gamma}}\ \vdash\ \underline{a}_{\gamma} : A_{\gamma}}$ $\natural$-variable rule	$$\begin{tikzcd} \Gamma' \arrow[rr, two heads, "p_{\Gamma'}"] \arrow[dr, dashed, "\vdash\underline{(-)}"] \arrow[dd, equal] & & \natural^{\text{rel}}_{\Gamma} A \arrow[rr] \arrow[dd, two heads] \arrow[drr, phantom, "\text{(pb)}"] & & \natural A \arrow[r, "\text{obt}^{\natural}_{A}"] \arrow[dd, "\natural p_A"] & A \arrow[dd, two heads, "p_A"] \\ & A_{\Gamma'} \arrow[urrrr] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & & & \\ \Gamma' \arrow[r, equal] & \Gamma' \arrow[r, two heads] & \Gamma \arrow[r, "\text{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\text{ret}^{\natural}_{\Gamma}"] \arrow[rr, bend right, "\text{id}"', "\text{(99)}"] & \natural\natural\Gamma \arrow[r, "\text{obt}^{\natural}_{\natural\Gamma}"] & \natural\Gamma \end{tikzcd}$$	(109)
$\natural\text{VAR-PROJ}\ \dfrac{\gamma{:}\Gamma\ \vdash\ A_{\gamma} : \text{Type}}{\Gamma,\ x : A,\ \Gamma'\ \vdash\ \underline{x} : \underline{A}}$ projective $\natural$-variable rule	$$\begin{tikzcd} \Gamma' \arrow[rr, "p_{\Gamma'}"] \arrow[dr, dashed, "\vdash\underline{(-)}"] \arrow[dd, equal] & & A \arrow[r, "\text{ret}^{\natural}_{A}"] \arrow[dd, two heads, "p_A"] & \natural A \arrow[r, "\text{obt}^{\natural}_{\Gamma}"] \arrow[dd, two heads, "\natural p_A"] & A \arrow[dd, two heads, "p_A"] \\ & \underline{A}_{\Gamma'} \arrow[urrr] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & & & \\ \Gamma' \arrow[r, equal] & \Gamma' \arrow[r] & \Gamma \arrow[r, "\text{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\text{obt}^{\natural}_{\Gamma}"] & \Gamma \end{tikzcd}$$	(110)

$\natural$-Compatibility with function types. (Syntax according to [RFL21, Rem. 2.4])

$$\frac{\Gamma \;\vdash\; f : A \to B}{\underline{\Gamma} \;\vdash\; \underline{f} : \underline{A} \to \underline{B}}$$

$$\begin{tikzcd}
\natural\Gamma \arrow[rr, "\mathrm{obt}^\natural_\Gamma"] \arrow[dr, dashed, "\vdash \underline{f}"'] \arrow[dd, equal] & & \Gamma \arrow[r, "\vdash f"] \arrow[dd, equal] & \mathrm{Map}(p_A,\, p_B) \arrow[dd] & \\
& \underline{\mathrm{Map}(p_B,\, p_B)} \arrow[urr] \arrow[d, two heads] & & & \leftrightarrow \\
\natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\mathrm{obt}^\natural_\Gamma"] & \Gamma \arrow[r, equal] & \Gamma &
\end{tikzcd}$$

$$\begin{tikzcd}
\underline{A} \arrow[rr, "\widetilde{(\vdash f) \circ \mathrm{obt}^\natural_\Gamma}"] \arrow[dr, dashed, "\underline{f}"] \arrow[dd, two heads, "p_{\underline{A}} \equiv (\mathrm{obt}^\natural_\Gamma)^* p_A"'] & & B \arrow[dd, "p_B"] & & \natural\Gamma \arrow[r, dashed, "\vdash \underline{f}"] \arrow[dd, equal] & \mathrm{Map}(p_{\underline{A}},\, p_{\underline{B}}) \arrow[dd, two heads] \\
& \underline{B} \arrow[ur] \arrow[d, two heads, "p_{\underline{B}}"] \arrow[r, phantom, "\text{(pb)}"] & & \leftrightarrow & & \\
\natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\mathrm{obt}^\natural_\Gamma"] & \Gamma & & \natural\Gamma \arrow[r, equal] & \natural\Gamma
\end{tikzcd}$$

(111)

Inference rules for $\natural$. (Syntax from [RFL21, Fig. 2][Ri22, Fig. 1.2]).

Syntax	Semantics	
$\natural$-FORM $\dfrac{\underline{\Gamma} \ \vdash \ A : \mathrm{Type}}{\Gamma \ \vdash \ \natural A : \mathrm{Type}}$	$$\dfrac{\begin{tikzcd} A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \end{tikzcd}}{\begin{tikzcd} \text{(88)} & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural(\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[r, "\natural q_A"] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}}$$	(112)
$\natural$-INTRO $\dfrac{\underline{\Gamma} \ \vdash \ a : A}{\Gamma \ \vdash \ a^{\natural} : \natural A}$	$$\dfrac{\begin{tikzcd} \natural\Gamma \arrow[r, "\vdash a"] \arrow[d, equal] & A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \arrow[r, equal] & \natural\Gamma \end{tikzcd}}{\begin{tikzcd} \Gamma \arrow[rrr, "\mathrm{ret}^{\natural}_{\Gamma}"] \arrow[dr, dashed] \arrow[ddrr, dashed, bend right, "\vdash a^{\natural}"'] \arrow[ddd, equal] & & & \natural\Gamma \arrow[r, "\vdash a"] \arrow[ddd, equal] & A \arrow[r, "\mathrm{ret}^{\natural}_{A}"] \arrow[ddd, two heads, "p_A"] & \natural A \arrow[ddd, two heads, "\natural p_A"] \\ & (\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[urrr, "q_A"] \arrow[dd] \arrow[dr, dashed, "\mathrm{ret}^{\natural^{\mathrm{rel}}_{\Gamma}}_{A}"] & \text{(91)} & & & \\ & & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[uurrr] \arrow[d] & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\natural\Gamma}"] & \natural\natural\Gamma \end{tikzcd}}$$	(113)
$\natural$-ELIM $\dfrac{\Gamma \ \vdash \ b : \natural A}{\Gamma \ \vdash \ b_{\natural} : A}$	$$\dfrac{\begin{tikzcd} \Gamma \arrow[r, "\vdash b"] \arrow[d, equal] & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural\big((\mathrm{ret}^{\natural}_{\Gamma})^* A\big) \arrow[r, "\natural q_A"] \arrow[d, two heads] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, two heads, "\natural p_A"] \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}}{\begin{tikzcd} \Gamma \arrow[rr, "\vdash b"] \arrow[dr, dashed, "b_{\natural}"'] \arrow[dd, equal] & & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[dd, two heads] \arrow[ddr, phantom, "\text{(pb)}"] & \natural\big((\mathrm{ret}^{\natural}_{\Gamma})^* A\big) \arrow[r] \arrow[dd, two heads] \arrow[ddr, phantom, "\text{(pb)}"] & \natural A \arrow[r, "\mathrm{obt}^{\natural}_{A}"] \arrow[dd, two heads, "\natural p_A"] & A \arrow[dd, two heads, "p_A"] \\ & (\mathrm{ret}^{\natural}_{\Gamma})^* A \arrow[urrrr] \arrow[d] & & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \arrow[r, "\natural\mathrm{ret}^{\natural}_{\Gamma}"] \arrow[rr, bend right, "\mathrm{id}"', "\text{(99)}"] & \natural\natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\natural\Gamma}"] & \natural\Gamma \end{tikzcd}}$$	(114)
$\natural$-BETA $\dfrac{\underline{\Gamma} \ \vdash \ a : A}{\Gamma \ \vdash \ {a^{\natural}}_{\natural} \equiv a : A}$	$$\dfrac{\begin{tikzcd} \natural\Gamma \arrow[r, "\mathtt{bind}^{\natural}_{(\vdash a)}"] \arrow[rr, bend left, "\vdash a"] \arrow[d, equal] & \natural A \arrow[r, "\mathrm{obt}^{\natural}_{A}"] & A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \arrow[rr, equal] & & \natural\Gamma \end{tikzcd}}{\begin{tikzcd} \natural\Gamma \arrow[r, "\mathtt{bind}^{\natural}_{(\vdash a)}"] \arrow[rrrr, bend left, "\vdash a"] \arrow[d, equal] & \natural A \arrow[r, "\mathrm{obt}^{\natural}_{A}"] \arrow[rr, bend left, "\mathrm{id}", "\text{(92)}"'] & A \arrow[r, "\mathrm{ret}^{\natural}_{A}"] \arrow[d, two heads, "p_A"] & \natural A \arrow[r, "\mathrm{obt}^{\natural}_{A}"] \arrow[d, two heads, "\natural p_A"] & A \arrow[d, two heads, "p_A"] \\ \natural\Gamma \arrow[rr, equal] & & \natural\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\natural\Gamma}"] \arrow[rr, bend right, "\mathrm{id}"', "\text{(99)}"] & \natural\natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\natural\Gamma}"] & \natural\Gamma \end{tikzcd}}$$	(115)

$\natural\text{-ETA}\ \dfrac{\Gamma \ \vdash \ b : \natural A}{\Gamma \ \vdash \ b \equiv \underline{b_{\natural}}^{\natural} : \natural A}$	$$\begin{tikzcd} \Gamma \arrow[r, "\vdash b"] \arrow[d, equal] & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[d, ->>, "\natural p_A"] \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\natural(\mathrm{ret}^{\natural}_{\Gamma}) \circ \mathrm{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \end{tikzcd}$$ $$\begin{tikzcd} \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"'] \arrow[d, equal] \arrow[rrrr, bend left, "\vdash b"] & \natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\Gamma}"'] \arrow[d] & \Gamma \arrow[r, "(101)", "\vdash b"'] \arrow[d, equal] & \natural^{\mathrm{rel}}_{\Gamma} A \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \natural A \arrow[r, "\mathrm{obt}^{\natural}_{A}"'] \arrow[d, ->>, "\natural p_A"] \arrow[rr, bend left, "\mathrm{id}"] & A \arrow[r, "(92)", "\mathrm{ret}^{\natural}_{A}"'] \arrow[d, ->>, "p_A"] & \natural A \arrow[d, ->>, "\natural p_A"] \\ \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] \arrow[rrrr, bend right, "\natural\mathrm{ret}^{\natural}_{\Gamma} \circ \mathrm{ret}^{\natural}_{\Gamma}"'] & \natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\Gamma}"] & \Gamma \arrow[r, equal, "(101)"'] & \Gamma \arrow[r, "\natural(\mathrm{ret}^{\natural}_{\Gamma}) \circ \mathrm{ret}^{\natural}_{\Gamma}"] & \natural\natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\natural\Gamma}"] \arrow[rr, bend right, "\mathrm{id}"'] & \natural\Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\natural\Gamma}", "(92)"'] & \natural\natural\Gamma \end{tikzcd}$$	(116)

Observe that $a \mapsto \underline{a}^{\natural}$ is now interpreted simply by postcomposition with the naturality square for $\mathrm{ret}^{\natural}$:

Internal construction of ♮-unit. (Syntax from [RFL21, Def. 2.1][Ri22, Def. 1.1.3])

Syntax	**Semantics**	
$\begin{array}{rc} & \Gamma \ \vdash \ a{:}A \\ \cline{2-2} \scriptstyle(107) & \underline{\Gamma} \ \vdash \ \underline{a}{:}\underline{A} \\ \cline{2-2} \scriptstyle(113) & \Gamma \ \vdash \ \underline{a}^{\natural}{:}\natural\underline{A} \end{array}$	$$\begin{tikzcd} \Gamma \arrow[rrr, "\mathrm{ret}^{\natural}_{\Gamma}" description] \arrow[rrrrrrr, bend left, "\mathrm{ret}^{\natural}_{A} \circ (\vdash a)"] \arrow[dr, dashed, "\vdash \underline{a}"] \arrow[ddrr, dashed, bend right, "\vdash \underline{a}^{\natural}"'] \arrow[ddd, equal] & & & \natural\Gamma \arrow[rr, "\mathrm{obt}^{\natural}_{\Gamma}" description, "(93)"] \arrow[dr, dashed, "\vdash \underline{a}"] \arrow[ddd, equal] & & \Gamma \arrow[r, "\vdash a" description] \arrow[ddd, equal] & A \arrow[r, "\mathrm{ret}^{\natural}_{A}" description] \arrow[ddd, ->>, "p_A"] & \natural A \arrow[r, "(\natural v_A)^{-1}", "\sim"'] \arrow[ddd, "\natural p_A"] & \natural\underline{A} \arrow[ddd, "\natural p_{\underline{A}}"] \\ & (\mathrm{ret}^{\natural}_{\Gamma})^{*}\underline{A} \arrow[rrr] \arrow[dr, dashed, "\mathrm{ret}^{\natural^{\mathrm{rel}}_{\Gamma}}_{\underline{A}}"] \arrow[dd, ->>] & & & \underline{A} \arrow[urr, "v_A"] \arrow[urrrr, bend right, "\mathrm{ret}^{\natural}_{\underline{A}}" description, "(95)"] \arrow[dd, ->>, "p_{\underline{A}}"] & & & & \\ & & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[uurrrrrr, bend right, "(100)"'] \arrow[d, ->>] \arrow[dr, phantom, "\text{(pb)}"] & & & & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"'] & \natural\Gamma \arrow[r, equal] & \natural\Gamma \arrow[r, "\mathrm{obt}^{\natural}_{\Gamma}" description] \arrow[rrr, bend right, "\mathrm{id}"', "(92)"] & \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}" description] & \natural\Gamma \arrow[r, "\sim", "\mathrm{ret}^{\natural}_{\natural\Gamma}"'] & \natural\natural\Gamma \end{tikzcd}$$	(117)
$\begin{array}{rc} & \Gamma \quad \vdash \quad A : \mathrm{Type} \\ \cline{2-2} \scriptstyle(102) & \Gamma,\ a{:}A \ \vdash \quad a : A \\ \cline{2-2} \scriptstyle(117) & \Gamma,\ a{:}A \ \vdash \quad \underline{a}^{\natural} : \underline{A} \\ \cline{2-2} & \Gamma \quad \vdash \ (a \mapsto \underline{a}^{\natural}) : A \to \natural\underline{A} \end{array}$	$$\begin{tikzcd} A \arrow[rr, "\mathrm{ret}^{\natural}_{A}" description] \arrow[dr, dashed, "a \mapsto \underline{a}^{\natural}"] \arrow[dd, ->>, "p_A"'] & & \natural A \arrow[dd, ->>, "\natural p_A"] \\ & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[ur] \arrow[d, ->>] \arrow[dr, phantom, "\text{(pb)}"] & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}" description] & \natural\Gamma \end{tikzcd}$$	(118)

♮-Compatibility with dependent pairs. (Syntax from [RFL21, Prop. 2.16][Ri22, Prop. 1.1.18]):

Syntax	**Semantics**	
$\begin{array}{c} \Gamma \ \vdash \quad A : \mathrm{Type} \\ \Gamma,\ A \vdash \quad B : \mathrm{Type} \\ \hline \Gamma \ \vdash \natural\big((x : \underline{A}) \times \underline{B}(x)\big) \\ \simeq (u{:}\natural\underline{A}) \times \natural\underline{B}(\underline{u}_{\natural}) \end{array}$	$$\begin{tikzcd} & \natural^{\mathrm{rel}}_{\natural^{\mathrm{rel}}_{\Gamma}\underline{A}} \underline{B}_{\underline{(-)}_{\natural}} \arrow[r] & \natural B \arrow[rr] \arrow[dd, ->>, "\natural p_B"] \arrow[drr, phantom, "\text{(pb)}"] & & \natural\big(B_{\underline{(-)}_{\natural}}\big) \arrow[rr] \arrow[dd, ->>] \arrow[drr, phantom, "(104)\,\&\,(114)\ \text{(pb)}"] & & \natural B \arrow[dd, ->>, "\natural p_B"] \\ \natural^{\mathrm{rel}}_{\Gamma}\underline{B} \arrow[ur, equal, "\sim", "(84)\,\&\,(118)"'] \arrow[d, ->>] \arrow[dr, phantom, "\text{(pb)}"] & & & & & & \\ \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[rr, "q_{\natural A}" description] \arrow[dd, ->>] \arrow[rrrr, bend right, "\mathrm{ret}^{\natural}_{\natural^{\mathrm{rel}}_{\Gamma}\underline{A}}" description, "(87)"] \arrow[ddrr, phantom, "\text{(pb)}"] & & \natural A \arrow[r, "\mathrm{ret}^{\natural}_{\natural A}" description] \arrow[dd, ->>, "\natural p_A"] \arrow[rrrr, bend left, "\mathrm{id}" description, "(99)"'] & \natural\natural A \arrow[r, "(\natural q_{\natural A})^{-1}"] & \natural\big(\natural^{\mathrm{rel}}_{\Gamma} A\big) \arrow[r, "\natural q_A"] & \natural\natural A \arrow[r, "\natural\mathrm{obt}^{\natural}_{A}" description] & \natural A \\ & & & & & & \\ \Gamma \arrow[rr, "\mathrm{ret}^{\natural}_{\Gamma}" description] & & \natural\Gamma & & & & \end{tikzcd}$$	(119)

3.2.3 Syntactic representation of the Motivic Yoga

We turn to the construction of dependent linear types, denoted QuType_W in §3.1.

We show 1-categorical semantics (identity types are interpreted as diagonal maps $\Delta_A : A \to A \times A$).

Linear types. (Syntax from [RFL21, pp. 24][Ri22, §2.1])

Syntax	Semantics	
$\dfrac{\Gamma \;\vdash\; A{:}\mathrm{Type} \qquad \Gamma \;\vdash\; \underline{x}{:}\natural\underline{A}}{\Gamma \;\vdash\; A_{\underline{x}} \;:\equiv\; (a{:}A) \times \mathrm{Id}(\underline{a}^{\natural}, \underline{x})}$ linear fiber	$$\begin{tikzcd} A_{\underline{x}} \arrow[rrr, "(\mathrm{pb})"'] \arrow[dr, "(pb)"'] \arrow[dd, ->>] \arrow[dddd, ->>, bend right, "p_{A_{\underline{x}}}"'] & & & \natural\Gamma \arrow[r, "\vdash\underline{x}", "(\mathrm{pb})"'] \arrow[dd, ->>, "(\vdash\underline{x}, \mathrm{id})"] & \natural A \arrow[dd, ->>, "\Delta_A"] \\ & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[urrr, "(\mathrm{pb})"'] \arrow[dd, ->>, "\Delta_{\natural^{\mathrm{rel}}_{\Gamma}\underline{A}}"] & & & \\ A \arrow[rrr, "\big(\mathrm{ret}^{\natural}_{A},\, \mathrm{ret}^{\natural}_{\Gamma} \circ p_A\big)"'] \arrow[dr, "\vdash\underline{((-)^{\natural}}, \underline{x})", "(118)"'] \arrow[dd, ->>, "p_A"] & & & \natural A \times \natural\Gamma \arrow[r, "\mathrm{id}\times\vdash\underline{x}"] & \natural A \times \natural A \arrow[dd, ->>, "\natural p_A \times \natural p_A"] \\ & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \times_{\Gamma} \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[urrr, "(\mathrm{pb})"'] \arrow[d, ->>] & & & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[rr, "\mathrm{ret}^{\natural}_{\Gamma}"] & & \natural\Gamma \arrow[r, "\Delta_{\natural\Gamma}"] & \natural\Gamma \times \natural\Gamma \end{tikzcd}$$	(120)
$\dfrac{\Gamma \;\vdash\; A{:}\mathrm{Type} \qquad \Gamma \;\vdash\; \underline{x}{:}\natural\underline{A}}{\Gamma \;\vdash\; \natural\big(A_{\underline{x}}\big) \;\simeq\; *}$ linear fibers are indeed linear	$$\begin{tikzcd} \natural A_{\underline{x}} \arrow[r, equal, "\sim", "(120)"'] & \natural\Gamma \arrow[r, "\natural\mathrm{ret}^{\natural}_{\Gamma}", "(97)"'] \arrow[d, ->>, "\vdash\underline{x}"] \arrow[dr, phantom, "(\mathrm{pb})"] & \natural\natural\Gamma \arrow[d, ->>, "\natural(\vdash\underline{x}, \mathrm{id})"] \\ & \natural A \arrow[r, "\natural\big(\mathrm{ret}^{\natural}_{A},\, \mathrm{ret}^{\natural}_{A} \circ p_A\big)"] & \natural\big(\natural A \times \natural\Gamma\big) \end{tikzcd}$$	(121)
$\dfrac{\Gamma \quad \vdash \qquad A : \mathrm{Type}}{\Gamma \quad \vdash \quad A \;\simeq\; (x{:}\natural\underline{A}) \times A_{\underline{x}}}$ types are sums of their linear fibers	$$\begin{tikzcd} A \arrow[rr, "\mathrm{ret}^{\natural}_{A}"] \arrow[dr, "(\mathrm{pb})", "(pb)"'] \arrow[dd, ->>, "(\mathrm{id}_A, \mathrm{ret}^{\natural}_{A})"] \arrow[dddd, ->>, bend right, "p_A"'] & & \natural A \arrow[dd, ->>, "\Delta_{\natural A}"] \\ & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[ur, "(\mathrm{pb})"'] \arrow[dd, ->>] & \\ A \times_{\Gamma} \natural A \arrow[rr, "(\mathrm{ret}^{\natural}_{A}, \mathrm{id}_{\natural A})"] \arrow[dr] \arrow[dd, ->>] & & \natural A \times_{\Gamma} \natural A \arrow[dd, ->>] \\ & \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \times_{\Gamma} \natural^{\mathrm{rel}}_{\Gamma}\underline{A} \arrow[ur, "(\mathrm{pb})"'] \arrow[d, ->>] & \\ \Gamma \arrow[r, equal] & \Gamma \arrow[r, "\mathrm{ret}^{\natural}_{\Gamma}"] & \natural\Gamma \end{tikzcd}$$	(122)

(...)

Conclusion. There exists an extension `LHoTT` of classical `HoTT` (Lit. 2.6) which serves as the internal logic for categories of linear bundle types as in §3.1 and in [EoS], in particular reflecting the *Motivic Yoga* of operations on such categories. Using this linear homotopy type theory, all of the quantum language constructions which we consider in the following can in principle be encoded, i.e. these quantum language constructs are just *syntactic sugar* for `LHoTT` code. That said, here we will not further dwell onformal `LHoTT`, the reader may find example translations discussed in [Ri23].

4 Quantum effects

We show that a system of basic (co)monads which is canonically *defineable* in dependent linear homotopy type theory (`LHoTT`) equips the underlying (independent) linear type theory with the computational effects which otherwise have to be postulated in (typed) quantum programming languages: besides a quantization modality ($\natural$) (turning bits into q-bits, etc.), these effects notably include quantum measurement ($\bigcirc$) and conditional quantum state preparation ($\star$), which turn out to correspond to Coecke et al.'s "classical structures" Frobenius monad.

§4.1 – Classical epistemic logic via Dependent classical types;
§4.2 – Quantum epistemic logic via Dependent linear types;
§4.3 – Controlled quantum gates via Quantum effect logic.

4.1 Classical epistemic logic via Dependent types

We lay out our perspective (following [nLab14][Co20, Ch. 4]) on (S5 Kripke semantics for) modal logic/type theory (Lit. 2.13). This is naturally realized (see Rem. 4.4 below) by *dependent* type theory (Lit. 2.4), with "possible worlds" given by terms of base types and with modal operators given by the (co)monads induced by dependent (co)product[18] type formation followed by context re-extension. This discussion is to prepare the ground for our formal quantum epistemic logic in §4.2.

For expository convenience, we speak in the 1-categorical semantics where the type universe "ClaType" refers to a topos of types (e.g.: Set) and for B : Type the universe $\mathrm{ClaType}_B$ of B-dependent types refers to the slice topos over B. All of the discussion is readily adapted to homotopy type theory proper and its ∞-topos semantics without any relevant changes, whence we do not dwell on it here (the homotopy theoretic aspect does become relevant further below). The crux is that all the constructions considered now are readily available inside a dependently typed language such as `HoTT` or `LHoTT`.

Dependent type formation by base change. The starting point is the basic fact that any $W : \mathrm{Type}_\Gamma$, hence any *display map* $p_W : W \to \Gamma$, induces a *base change adjoint triple* between W-dependent types and bare types in the default context Γ:

$$\begin{tikzcd}[column sep=huge]
 & W \arrow[rr, "p_w"] & & \Gamma & \\
\textbf{W-dependent types} & \mathrm{ClaType}_W \arrow[rr, shift left=4, "\coprod_W", "\textbf{dependent co-product}"'] \arrow[rr, shift right=4, "\prod_W"', "\textbf{dependent product}"] \arrow[rr, phantom, shift left=2, "\perp"] \arrow[rr, phantom, shift right=2, "\perp"] & & \mathrm{ClaType}_\Gamma \arrow[ll, "\textbf{context} \;\; \times W \;\; \textbf{extension}"] & \textbf{types in default context}
\end{tikzcd} \tag{123}$$

via

$$\begin{array}{lclcl}
D : \mathrm{Type}_W & \vdash & \coprod_W D : & \Gamma \longrightarrow \mathrm{ClaType} \\
 & & & \gamma \longmapsto \coprod\limits_{w : \mathrm{fib}_\gamma(p_W)} D_w \\[1em]
D : \mathrm{Type}_\Gamma & \vdash & D \times W : & W \longrightarrow \mathrm{ClaType} \\
 & & & w \longmapsto D_{p_W(w)} \\[1em]
D : \mathrm{Type}_W & \vdash & \prod_W D : & \Gamma \longrightarrow \mathrm{ClaType} \\
 & & & \gamma \longmapsto \prod\limits_{w : \mathrm{fib}_\gamma(p_W)} D_w
\end{array} \tag{124}$$

whose (co)restriction along

$$\begin{tikzcd}[column sep=large]
\textbf{types} & \mathrm{ClaType}_\Gamma \arrow[r, shift left=1.5, "{[\text{-}]_0}", "\textbf{propositional truncation}"'] \arrow[r, phantom, "\perp"] & \mathrm{Prop}_\Gamma \arrow[l, hook, shift left=1.5] & \textbf{propositions}
\end{tikzcd} \tag{125}$$

gives the quantifiers of first-order logic:

$$\begin{tikzcd}[column sep=huge]
\textbf{W-dependent propositions} & \mathrm{Prop}_W \arrow[rr, shift left=4, "{\exists_W = \left[\coprod_W(\text{-})\right]_0}", "\textbf{existential quantification}"'] \arrow[rr, shift right=4, "\forall_W = \prod_W"', "\textbf{universal quantification}"] \arrow[rr, phantom, shift left=2, "\perp"] \arrow[rr, phantom, shift right=2, "\perp"] & & \mathrm{Prop}_\Gamma \arrow[ll, "\textbf{context} \;\; \times W \;\; \textbf{extension}"] & \textbf{propositions in default context}
\end{tikzcd} \tag{126}$$

It is immediate (and generally well-known but has previously received little attention in modal type theory)

---

[18] We say *dependent co-product* "$\coprod_B$" for what is traditionally called the *dependent sum* "$\sum_B$" in intuitionistic type theory. Apart from being the more descriptive term, this avoids a clash of terminology after passage to *linear* type theory where actual linear sums of types ("direct sums") do play a(nother) role.

that by composing the adjoint type constructors (123) to endo-functors yields a pair of adjoint pairs of (co)monads:

$$\begin{tikzcd}[column sep=large]
 & W \arrow[rr, "p_W"] & & \Gamma & \\
\text{actual data} \arrow[r, phantom, "\bot"] & \mathrm{ClaType}_W \arrow[loop above, "\overset{\text{possibly}}{\Diamond_W}"] \arrow[loop below, "\underset{\text{necessarily}}{\Box_W}"'] \arrow[rr, shift left=4, "\overset{\text{dependent co-product}}{\coprod_W}"] \arrow[rr, shift right=4, "\underset{\text{dependent product}}{\prod_W}"'] & & \mathrm{ClaType}_\Gamma \arrow[ll, "\overset{\bot}{\underset{\bot}{(-)\times W}}" description] \arrow[loop above, "\overset{\text{randomly}}{\text{☆}_W}"] \arrow[loop below, "\underset{\text{indefinitely}}{\bigcirc_W}"'] \arrow[r, phantom, "\bot"] & \text{potential data}
\end{tikzcd}$$
(127)

whose (co)restriction along propositional truncation (125) we shall denote by the same symbols:

$$\begin{tikzcd}[column sep=large]
 & W \arrow[rr, "p_W"] & & \Gamma & \\
\text{actual propositions} \arrow[r, phantom, "\bot"] & \mathrm{Prop}_W \arrow[loop above, "\overset{\text{possibly}}{\Diamond_W}"] \arrow[loop below, "\underset{\text{necessarily}}{\Box_W}"'] \arrow[rr, shift left=4, "\overset{\substack{\text{0-truncated} \\ \text{dependent co-product}}}{\left[\coprod_W (-)\right]_0}"] \arrow[rr, shift right=4, "\underset{\text{dependent product}}{\prod_W}"'] & & \mathrm{Prop}_\Gamma \arrow[ll, "\overset{\bot}{\underset{\bot}{(-)\times W}}" description] \arrow[loop above, "\overset{\text{randomly}}{\text{☆}_W}"] \arrow[loop below, "\underset{\text{indefinitely}}{\bigcirc_W}"'] \arrow[r, phantom, "\bot"] & \text{potential propositions}
\end{tikzcd}$$
(128)

Actuality logic. The terminology on the left of diagram(127) is justified by the following Remark 4.1 and the observation of Theorem 4.3 below, which we articulate as a *theorem* not because its proof would be much more than an unwinding of definitions (nor surprising, in view of [Law69a]), but to highlight its Yoneda-Lemma-like conceptual importance:

Remark 4.1 (Epistemic interpretation of dependent types)**.** Concretely, we may read these modal operators (127) as follows, where we use the traditional language of "possible worlds" (Lit. 2.13) but suggest to think of these "worlds" quite concretely as classical states of an observed universe to the extent partially revealed by a particular measurement, hence like the "many worlds" of quantum epistemology (Lit. 2.2).

(i) Given a proposition $P_\bullet$ which depends on which world w is or has been measured:

$\Box_W P_\bullet$ means:	P_w means:	$\Diamond_W P_\bullet$ as:
"P_w *does* or *is known to* hold *necessarily*" namely, no matter which world w is measured.	"P_w *does* or *is known* to hold *actually*" namely for the *given* world w measured.	"P_w *does* or *is known* to hold *possibly*" namely for *some* possibly measured world w.

(ii) Moreover, the (co)unit $\mathrm{ret}^{\Diamond}$ ($\mathrm{obt}^{\Box}$) of the above (co)monads reflect the logical entailment of these modal propositions:

$$\begin{tikzcd}[column sep=large]
 & \overset{\text{necessarily } D_\bullet}{\Box_W D_\bullet} \arrow[r, "\overset{\text{entails}}{\mathrm{obt}^{\Box_W}_{D_\bullet}}"] & \overset{\text{actually } D_\bullet}{D_\bullet} \arrow[r, "\overset{\text{entails}}{\mathrm{ret}^{\Diamond_W}_{D_\bullet}}"] & \overset{\text{possibly } D_\bullet}{\Diamond_W D_\bullet} \\
w : W \quad \vdash & \prod\limits_{w':W} D_{w'} \arrow[r, "(d_{w':W}) \mapsto d_w"] & D_w \arrow[r, "d_w \mapsto (w, d_w)"] & \coprod\limits_{w':W} D_{w'}
\end{tikzcd}$$
(129)

Remark 4.2 (Hexagon of epistemic entailments)**.** The *naturality* of the transformations (129) is reflected in commuting squares as shown in the following diagram (130), whose hexagonal composition gives the diagram (7) announced in the Introduction (there evaluated for linear/quantum types, which we come to in §4.2, but the existence of the commuting hexagon as such depends only on the naturality of the epistemic entailments):

$$\begin{array}{l} D_\bullet,\ D'_\bullet \;:\; \mathrm{Type}_W \\ G_\bullet \;:\; D_\bullet \to D'_\bullet \end{array} \quad \vdash \tag{130}$$

$$\begin{tikzcd}
 & \Box\Diamond D_\bullet \arrow[rr, "\Box\Diamond G_\bullet"] & & \Box\Diamond D'_\bullet \arrow[dr, "\mathrm{obt}^{\Box}_{\Diamond D'_\bullet}"] & \\
 & \Box(\mathrm{ret}^{\Diamond}_{G_\bullet}) & & & \\
\Box D_\bullet \arrow[uur, "\Box\left(\mathrm{ret}^{\Diamond}_{D_\bullet}\right)"] \arrow[rr, "\Box G_\bullet"] \arrow[ddr, "\mathrm{obt}^{\Box}_{D_\bullet}"'] & & \Box D'_\bullet \arrow[uur, "\Box\left(\mathrm{ret}^{\Diamond}_{D'_\bullet}\right)"] \arrow[ddr, "\mathrm{obt}^{\Box}_{D'_\bullet}"'] & \mathrm{obt}^{\Box}_{(\mathrm{ret}^{\Diamond}_{D'_\bullet})} & \Diamond D'_\bullet \\
 & \mathrm{obt}^{\Box}_{G_\bullet} & & & \\
 & D_\bullet \arrow[rr, "G_\bullet"] & & D'_\bullet \arrow[uur, "\mathrm{ret}^{\Diamond}_{D'_\bullet}"'] &
\end{tikzcd}$$

$$\|$$

$$\begin{tikzcd}
 & \Box\Diamond D_\bullet \arrow[rr, "\Box\Diamond G_\bullet"] \arrow[dr, "\mathrm{obt}^{\Box}_{\Diamond D_\bullet}"] & & \Box\Diamond D'_\bullet \arrow[dr, "\mathrm{obt}^{\Box}_{\Diamond D'_\bullet}"] & \\
 & & & \mathrm{obt}^{\Box}_{\Diamond G_\bullet} & \\
\Box D_\bullet \arrow[uur, "\Box\left(\mathrm{ret}^{\Diamond}_{D_\bullet}\right)"] \arrow[ddr, "\mathrm{obt}^{\Box}_{D_\bullet}"'] & \mathrm{obt}^{\Box}_{(\mathrm{ret}^{\Diamond}_{D_\bullet})} & \Diamond D_\bullet \arrow[rr, "\Diamond G_\bullet"] & & \Diamond D'_\bullet \\
 & & & \mathrm{ret}^{\Diamond}_{G_\bullet} & \\
 & D_\bullet \arrow[rr, "G_\bullet"] \arrow[uur, "\mathrm{ret}^{\Diamond}_{D_\bullet}"'] & & D'_\bullet \arrow[uur, "\mathrm{ret}^{\Diamond}_{D'_\bullet}"'] &
\end{tikzcd}$$

For emphasis, the following theorem highlights that this epistemic logic of dependent types recovers what is traditionally understood in modal logic:

Theorem 4.3 (S5 Kripke semantics as co-monadic descent). *The possible-worlds Kripke semantics* (20) *for S5 modal logic are precisely given by dependent type formation* (127) *(for* ClaType $\equiv$ Set*) where a Kripke frame* $\big(W : \mathrm{Set},\, R : W \times W \to \mathrm{Prop}\big)$ *corresponds to that display map* (123) *which is its quotient projection* $p_W : W \twoheadrightarrow \Gamma \equiv W_{/R}$.

Proof. A classical theorem ([Kr63][FHMV95, Thm. 3.1.5], cf. [Sa10]) identifies the Kripke semantics for S5 modal logic with precisely those Kripke frames (W, R) where R is an equivalence relation. The equivalence classes Γ of R hence form a partition of W as

$$W \;=\; \coprod_{\gamma : \Gamma} \mathrm{fib}_\gamma(p_W)\,,$$

which gives the incarnation of W as a Γ-dependent type. By (124), the induced comonad (127) acts as

$$P : \mathrm{Prop}_W \qquad \vdash \qquad \begin{array}{rcl} \Box_W P \;:\; W & \longrightarrow & \mathrm{Prop} \\ w & \mapsto & \underset{w' \,:\, \mathrm{fib}_{p_W(w)}(p_W)}{\forall} P(w') \end{array} \tag{131}$$

But with p_W identified as the quotient coprojection of R, we have

$$\mathrm{fib}_{p_W(w)}(p_W) \;\;=\;\; (w' : W) \times R(w, w')$$

whence (131) equals the traditional formula (20) for the Kripke semantics of the modal operator. ☐

Remark 4.4 (Dependent type theory as universal Epistemic modal type theory)**.** Thm. 4.3 suggests that one may regard dependent type theory equivalently as a universal form of epistemic type theory (Lit. 2.14) in generalization of how modal logics may be viewed as an equivalent perspective on (fragments) of first-order logic (cf. [BvBW07, pp. xiii]). In both cases one switches perspective from type formation by base change adjoint triples (123)(126) to the associated adjoint pairs of (co)monads (127)(128). (An analogous change in perspective happens in (algebraic) geometry when expressing *descent theory* in terms of *monadic descent*.)

Noticing that the development of general modal type theory is still in its infancy with its general *linear* form hardly known at all, this change of perspective allows us to use (in §4.2) well-developed (linear) dependent type theory to realize the epistemic form of modal type theory that we need for certifying quantum protocols.

Potentiality logic. The (co)monads on the right side of (127) are known in effectful classical computer science (Lit. 2.15) as the *W-(co)reader (co)monad*, (48) often denoted as on the right here:

$$\begin{aligned} \bigcirc_W D &\equiv [W,\, D] && \textbf{W-reader monad} \\ \text{☆}_W D &\equiv W \times D && \textbf{W-coreader comonad} \end{aligned} \tag{132}$$

What has not previously found attention is the corresponding modal/epistemic perspective on these operators. It will be useful to dwell on this point a little. Our suggestion in (127) of *potentiality* as the antonym to *actuality* (the latter well-established in modal logic) follows Aristotle and Heisenberg (as recounted in [Ja17]). In further support of this nomenclature we offer the following fact, which gives a precise sense that:

Potential data is equivalently data whose possibility entails its actuality, consistently

$$\begin{array}{ccc} \mathrm{ClaType}_\Gamma & \xleftrightarrow{\qquad\qquad\qquad\sim\qquad\qquad\qquad} & \mathrm{ClaType}_W^{\Diamond_W} \\[6pt] \underset{\textbf{potential data}}{D : \mathrm{Type}_\Gamma} & \underset{\textbf{is equivalently}}{\xleftrightarrow{\;\;\sim\;\;}} & \Big(\underset{\textbf{data whose}}{D_\bullet : \mathrm{Type}_W},\ \underset{\textbf{possibility entails its actuality,}}{\rho : \Diamond_W D_\bullet \longrightarrow D_\bullet},\ \underset{\textbf{consistently}}{\mathrm{utl}_{\Diamond_W}(\rho),\ \mathrm{act}_{\Diamond_W}(\rho)}\Big) \end{array} \tag{133}$$

(This compares favorably with the traditional informal intention of the "potentiality" modality, cf. [FG16, §44].) Namely, we have:

Proposition 4.5 (Potential data as possibility modal data)**.** *For $p_W : W \twoheadrightarrow \Gamma$ an epimorphism (as in Thm. 4.3), the context extension $(\text{-}) \times W : \mathrm{ClaType}_\Gamma \to \mathrm{ClaType}_W$ is monadic* (45) *whence the potential types* (127) *are identified with the (free) possibility-modal types* (41) *and hence* (49) *also with the necessity-modal types:*

$$\begin{tikzcd} & \mathrm{ClaType}_W \arrow[loop above, "\Diamond_W\ \text{possibly}"] \arrow[loop below, "\Box_W\ \text{necessarily}"'] \arrow[r, phantom, ""] & & \mathrm{ClaType}_W^{\Diamond_W} & \textbf{possibility modal data} \\ \textbf{actual data}\ \perp & \mathrm{ClaType}_W \arrow[urr, "\coprod_W"] \arrow[drr, "\prod_W"'] & \arrow[u, phantom, "\perp"] \arrow[d, phantom, "\perp"] & \mathrm{ClaType}_\Gamma \arrow[ll, "\times W"'] \arrow[u, phantom, "\wr"] \arrow[d, phantom, "\wr"] & \textbf{potential data} \\ & & & \mathrm{ClaType}_W^{\Box_W} & \textbf{necessity modal data} \end{tikzcd} \tag{134}$$

Proof. By the Monadicity Theorem (45) and since the functor $(\text{-}) \times W$ has both a left and a right adjoint, it is sufficient to see that it reflects isomorphisms; but this follows immediately from the assumption that p_W is surjective. Compare to [Jo02, Lem. 1.3.2], namely if $(f \times W)_w \equiv f_{p_W(w)}$ is an isomorphism for $w : W$ then surjectivty of p_W implies that f_γ is an isomorphism for $\gamma : \Gamma$. $\square$

Remark 4.6 (Relation to monadic descent)**.** The statement and proof of Prop. 4.5 correspond to what in (algebraic) geometry is known as *monadic descent* (e.g. [JT94, §2.1]): In this context, the display map p_W would be called an *effective descent morphism*, and $\Diamond_W$-modale structure would be called *descent data* along p_W.

Remark 4.7 (Relation to lenses)**.** In the case Type = Set, the statement of Prop. 4.5 is known in the theory of *lenses* in computer science. Here one regards $\Diamond_W$-modale structure as a data base-type S equipped with functionality to read out (get) and to over-write (put) W-data subject to consistency conditions ("lawful lenses"):

$$\left(\begin{array}{cccc} \textbf{slice object} & \textbf{$\Diamond_W$-modale structure} & \textbf{$\Diamond_W$-unit law} & \textbf{$\Diamond_W$-action property} \\ \begin{bmatrix} S \\ \big\downarrow \scriptstyle\mathrm{get} \\ W \end{bmatrix} \in \mathrm{Type}_W & \begin{tikzcd} S \times W \arrow[rr, "\mathrm{put}"] \arrow[dr, "\mathrm{pr}_W"'] & & S \arrow[dl, "\mathrm{get}"] \\ & W & \end{tikzcd} & \begin{tikzcd} & W \times S \arrow[dr, "\mathrm{put}"] & \\ S \arrow[ur, "\mathrm{get}\times\mathrm{id}"] \arrow[rr, equal] & & S \end{tikzcd} & \begin{tikzcd} W \times W \times S \arrow[r, "\mathrm{pr}_1\times\mathrm{pr}_3"] \arrow[d, "\mathrm{id}_W\times\mathrm{put}"'] & W \times S \arrow[d, "\mathrm{put}"] \\ W \times S \arrow[r, "\mathrm{put}"] & S \end{tikzcd} \\ \textbf{database type S with} & \textbf{W-write functionality} & \textbf{overwriting identi-} & \textbf{subsequent writing} \\ \textbf{W-read functionality} & \textbf{verified by W-reading} & \textbf{cally has no effect} & \textbf{overwrites previous} \end{array} \right) : \big(\mathrm{Type}_W\big)^{\Diamond_W} \tag{135}$$

and the upshot of the monadicity statement (Prop. 4.5, [JRW10, Thm. 12][19]) is that this describes "addressed" access to a data sub-base type, in that such S are necessarily of product form $S \simeq W \times D$ with get $= \mathrm{pr}_W$, etc.

Random and (in)definite data. The (co)monads $\bigcirc$ ($\star$) on the right of (127) are well-known in terms of (co)effects in computer science (Lit. 2.15) as the "(co)reader (co)monad" (48), referring to the idea of a program *reading* (*providing*) a global variable $w : W$. However, for staying true to the spirit of modal logic, here we refer to these as the modalities of *indefiniteness* (*randomness*), in the following sense:

$\star_W D$ is the type of	D is the type of	$\bigcirc_W P_\bullet$ is the type of:
D-data d in a *definite* but *random* world w (as in "random access")	plain D-data d only *potentially* in some possible world	*indefinite* D-data $w \mapsto d_w$ contingent on a pending choice of possible world w.

$$\begin{tikzcd}[column sep=large] \textbf{randomly } P & \textbf{potentially } P & \textbf{indefinitely } P \\ \star_W P \arrow[r, "\mathrm{ret}_P^{\star_W}"', "\textbf{entails}"] & P \arrow[r, "\mathrm{obt}_P^{\bigcirc_W}"', "\textbf{entails}"] & \bigcirc_W P \\ \coprod\limits_{w':W} P \arrow[r, "(w,p)\mapsto p"] & P \arrow[r, "p \mapsto (w' \mapsto p)"] & \prod\limits_{w':W} P \end{tikzcd} \tag{136}$$

In particular, the monadic effect model (cf. Lit. 2.15) for operating on the parameter space W as on a *random access memory* (RAM) is the state monad (34), which we may realize as the composite

$$\underset{W}{\bigcirc}\,\underset{W}{\star}\, D \;\simeq\; \prod_W \coprod_W D \;\simeq\; \big[W,\, W \times D\big] \;\equiv\; W\mathrm{State}(D), \qquad \begin{tikzcd} \mathrm{Type} \arrow[loop left, "W\mathrm{State}"] \arrow[r, shift left=1.5, "\star_W"] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift left=1.5, "\bigcirc_W"] \end{tikzcd} . \tag{137}$$

It is in this common sense of *random access* as about "choice" (instead of "chance") that one should think about $\star_W$ as the modality of "being random".

In summary so far, we have found that any classical (intuitionistic) dependently typed language may be regarded as a rich epistemic modal type theory with, for every inhabited type W (in any ambient context Γ), the following identifications:

$$\begin{tikzcd}[column sep=large] \textbf{actual data} \; \perp & \mathrm{ClaType}_W \arrow[loop above, "\Diamond_W\ \textbf{possibly}"] \arrow[loop below, "\Box_W\ \textbf{necessarily}"'] \arrow[r, shift left=3, "\coprod_W", "\textbf{dependent co-product}"'{yshift=18pt}] \arrow[r, shift right=3, "\prod_W"', "\textbf{dependent product}"{yshift=-18pt}] \arrow[r, phantom, shift left=1.5, "\perp"] \arrow[r, phantom, shift right=1.5, "\perp"] & \mathrm{ClaType} \arrow[l, "\times W"'] \arrow[loop above, "\star_W\ \textbf{randomly}"] \arrow[loop below, "\bigcirc_W\ \textbf{indefinitely}"'] & \perp \; \textbf{potential data} \end{tikzcd}$$

$$\begin{tikzcd}[column sep=large] & \textbf{necessarily } P_\bullet & \textbf{actually } P_\bullet & \textbf{possibly } P_\bullet \\ & \Box_W P_\bullet \arrow[r, "\epsilon_{P_\bullet}^{\Box_W}"', "\textbf{entails}"] & P_\bullet \arrow[r, "\eta_{P_\bullet}^{\Diamond_W}"', "\textbf{entails}"] & \Diamond_W P_\bullet \\ {\color{red}w} : W \;\vdash & \prod\limits_{w':W} P_{w'} \arrow[r, "(w' \mapsto p_{w'}) \,\mapsto\, p_{\color{red}w}"] & P_{\color{red}w} \arrow[r, "p_{\color{red}w} \,\mapsto\, ({\color{red}w}, p_{\color{red}w})"] & \coprod\limits_{w':W} P_{w'} \\ & \textbf{randomly } P & \textbf{potentially } P & \textbf{indefinitely } P \\ & \star_W P \arrow[r, "\mathrm{ret}_P^{\star_W}"', "\textbf{entails}"] & P \arrow[r, "\mathrm{obt}_P^{\bigcirc_W}"', "\textbf{entails}"] & \bigcirc_W P \\ & \coprod\limits_{w':W} P \arrow[r, "(w,p)\mapsto p"] & P \arrow[r, "p \mapsto (w' \mapsto p)"] & \prod\limits_{w':W} P \end{tikzcd} \tag{138}$$

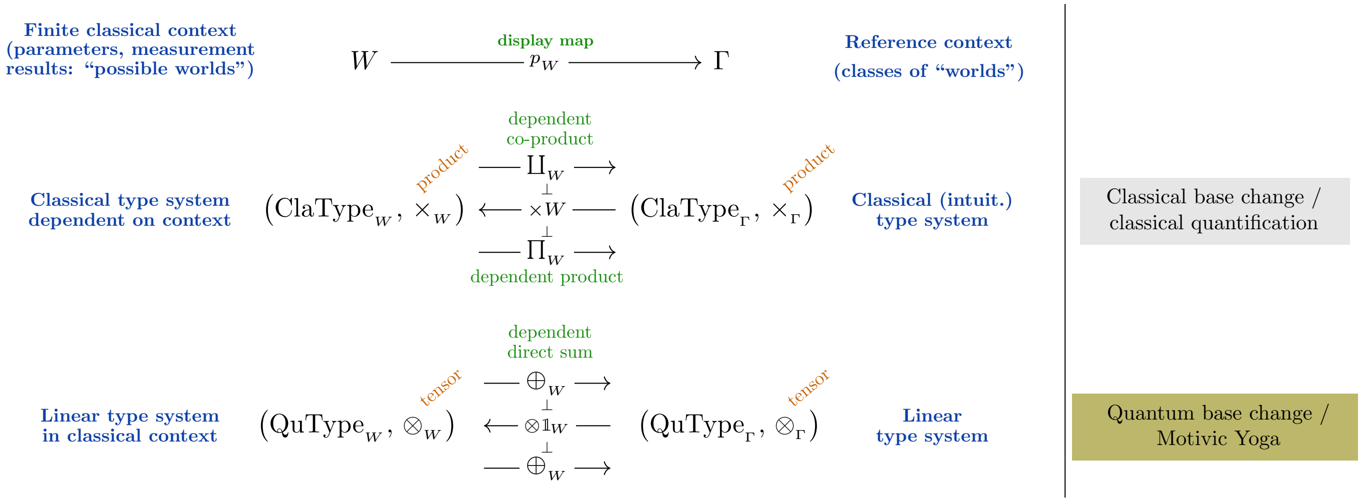
Next we proceed to find the quantum analog (142) of this logic.

---

[19] [Spi19] concludes from this situation that the theory of "lenses" is best regarded as an aspect of the much broader and classical theory of indexed categories (Grothendieck fibrations). Syntactically this means to regard them as an aspect of the theory of dependent types which – when also taking into account the related system of (co)monads – is the thesis that we are developing here.

4.2 Quantum epistemic logic via dependent linear types

On the backdrop (§4.1) of classical (intuitionistic) epistemic type theory understood as an equivalent re-interpretation of classical (intuitionistic) dependent type theory, and in view (§3) of the existence of dependent *linear* type theory `LHoTT`, we are led to expect that *quantum epistemic type theory* ought to analogously be obtained by re-regarding the base change adjunction (70) of dependent *linear* type formation



by passing to the induced (co)monads, which we denote by the same symbols as their classical counterparts (127):

$$\begin{tikzcd}
 & W \arrow[rr, "p_W"] & & \Gamma & \\
\text{Actual quantum data} \arrow[r, phantom, "\perp"] & \mathrm{QuType}_W \arrow[loop above, "\Diamond_W \text{ possibly}"] \arrow[loop below, "\Box_W \text{ necessarily}"'] \arrow[rr, shift left=3, "\text{dependent direct sum} \; \bigoplus_W"] \arrow[rr, shift right=3, "\bigoplus_W \; \text{dependent direct sum}"'] & & \mathrm{QuType}_\Gamma \arrow[ll, "\otimes \mathbb{1}_W" description] \arrow[loop above, "\text{randomly} \; \star_W"] \arrow[loop below, "\bigcirc_W \; \text{indefinitely}"'] \arrow[r, phantom, "\perp"] & \text{Potential quantum data}
\end{tikzcd}$$

(139)

A key point now is the *ambitexterity* (70) of the base change for dependent linear types along a finite classical type W:

$$W : \mathrm{FinClaType} \qquad \vdash \qquad \left(\bigoplus_W \dashv \otimes \mathbb{1}_W \dashv \bigoplus_W \right) \tag{140}$$

It is now as elementary to work out the (co)units of these (co)monads (they are the universal maps of the direct sum construction) as it is interesting – in view of quantum epistemology (Lit. 2.1):

Proposition 4.8. *The (co)units and (co)joins of the (co)monads in* (139) *are given, in components, as follows:*

Epistemic entailments in Quantum modal logic

$$\begin{tikzcd} \underset{W}{\square}\mathcal{H}_\bullet \arrow[r, "\mathrm{obt}^{\square_W}_{\mathcal{H}_\bullet}", "\text{necessity counit}"'] & \mathcal{H}_\bullet \end{tikzcd}$$

$$\begin{tikzcd} w : W \ \vdash \ \bigoplus_{w'}\mathcal{H}_w \arrow[r, "\oplus_{w'}|\psi_{w'}\rangle \ \mapsto \ |\psi_w\rangle", "\text{quantum state collapse}"'] & \mathcal{H}_w \end{tikzcd}$$

“ *what is necessary is actualized* ”

$$\begin{tikzcd} \mathcal{H}_\bullet \arrow[r, "\mathrm{ret}^{\lozenge_W}_{\mathcal{H}_\bullet}", "\text{possibility unit}"'] & \underset{W}{\lozenge}\mathcal{H}_\bullet \end{tikzcd}$$

$$\begin{tikzcd} w : W \ \vdash \ \mathcal{H}_w \arrow[r, "|\psi_w\rangle \ \mapsto \ \oplus_{w'}\delta^{w'}_{w}|\psi_w\rangle", "\text{quantum state preparation}"'] & \bigoplus_{w'}\mathcal{H}_{w'} \end{tikzcd}$$

“ *what is actual is possible* ”

“ *what is random exists potentially* ”

$$\begin{tikzcd} \underset{W}{\star}\mathcal{H} \arrow[r, hook, "\mathrm{obt}^{\star_W}_{\mathcal{H}}", "\text{randomness counit}"'] & \mathcal{H} \end{tikzcd}$$

$$\begin{tikzcd} \underset{w}{\bigoplus}\mathcal{H} \arrow[r, "\oplus_w|\psi_w\rangle \ \mapsto \ \sum_w |\psi_w\rangle", "\text{quantum superposition}"'] & \mathcal{H} \end{tikzcd}$$

“ *what exists potentially is indeterminate* ”

$$\begin{tikzcd} \mathcal{H} \arrow[r, "\mathrm{ret}^{\bigcirc_B}_{\mathcal{H}}", "\text{indefiniteness unit}"'] & \underset{W}{\bigcirc}\mathcal{H} \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{H} \arrow[r, "|\psi\rangle \ \mapsto \ \oplus_w|\psi\rangle", "\text{quantum parallelism}"'] & \underset{W}{\bigoplus}\mathcal{H} \end{tikzcd}$$

$$\begin{tikzcd} \underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}}", "\text{indefiniteness join}"'] & \underset{W}{\bigcirc}\mathcal{H} \\ \underset{w}{\bigoplus}\Big(\underset{W}{\square}\mathcal{H} \arrow[r, "\mathrm{obt}^{\square_W}_{\mathcal{H}}", "\text{quantum state collapse}"'] & \mathcal{H} \Big) \\ \oplus_{w'}|\psi_{w,w'}\rangle \arrow[r, mapsto] & |\psi_{w,w}\rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{W}{\star}\mathcal{H} \arrow[r, "\mathrm{dplc}^{\star_W}_{\mathcal{H}}", "\text{randomness cojoin}"'] & \underset{W}{\star}\underset{W}{\star}\mathcal{H} \\ \underset{w}{\bigoplus}\Big(\mathcal{H} \arrow[r, "\mathrm{obt}^{\lozenge_W}_{\mathcal{H}}", "\text{quantum state prepar.}"'] & \underset{W}{\lozenge}\mathcal{H} \Big) \\ |\psi_w\rangle \arrow[r, mapsto] & \oplus_{w'}\delta^{w'}_{w}|\psi_w\rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{W}{\lozenge}\underset{W}{\lozenge}\mathcal{H}_\bullet \arrow[r, "\mathrm{join}^{\lozenge_W}_{\mathcal{H}_\bullet}", "\text{possibility join}"'] & \underset{W}{\lozenge}\mathcal{H}_\bullet \\ w : W \ \vdash \ \underset{W}{\star}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathrm{obt}^{\star_W}_{\oplus_W\mathcal{H}_\bullet}", "\text{quantum superposition}"'] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \\ \underset{w''}{\bigoplus}|\psi_{w,w',w''}\rangle \arrow[r, mapsto] & \sum_{w''}|\psi_{w,w',w''}\rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{W}{\square}\mathcal{H}_\bullet \arrow[r, "\mathrm{dplc}^{\square_W}_{\mathcal{H}_\bullet}", "\text{necessity cojoin}"'] & \underset{W}{\square}\underset{W}{\square}\mathcal{H}_\bullet \\ w : A \ \vdash \ \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathrm{ret}^{\bigcirc_W}_{\oplus_W\mathcal{H}_\bullet}", "\text{quantum parallel.}"'] & \underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{H}_\bullet \\ |\psi_{w,w'}\rangle \arrow[r, mapsto] & \underset{w''}{\bigoplus}|\psi_{w,w'}\rangle \end{tikzcd}$$

Here the (co)joins in the lower half follow from the (co)units in the top half via (27).

Monadicity of quantum data. We observe that quantum data as in (139) is characterized by two monadicity theorems:

- Prop. 4.9: Potential quantum data is possibility-modal actual data.
- Prop. 4.11: Actual quantum data is indefiniteness-modal potential data.

First, we have the following quantum analog of the classical situation from Prop. 4.5:

Proposition 4.9 (Potential quantum data as possibility-modal actual data)**.** *For $p_W : W \twoheadrightarrow \Gamma$ an epimorphism (as in Thm. 4.3) the context extension $(\text{-}) \otimes \mathbb{1}_W : \mathrm{QuType}_\Gamma \to \mathrm{QuType}_W$ is monadic* (45) *whence the potential quantum types* (139) *are identified with the (free) possibility/necessity modal types* (41) *(just as classically* (134)*):*

$$\begin{tikzcd}
 & \overset{\text{possibly}}{\Diamond_W} \arrow[d, bend left] & {} \arrow[r, "\oplus_W"] \arrow[d, phantom, "\bot"] & \mathrm{QuType}_W^{\Diamond_W} \arrow[d, phantom, "\wr"] & \text{possibility modal data} \\
\text{actual quantum data} \;\; \bot & \mathrm{QuType}_W & & \mathrm{QuType}_\Gamma \arrow[ll, "\otimes \mathbb{1}_W"'] & \text{potential quantum data} \\
 & \underset{\text{necessarily}}{\Box_W} \arrow[u, bend right] & {} \arrow[r, "\oplus_W"'] \arrow[u, phantom, "\bot"] & \mathrm{QuType}_W^{\Box_W} \arrow[u, phantom, "\wr"] & \text{necessity modal data}
\end{tikzcd} \tag{141}$$

Proof. This statement has verbatim the same abstract proof – via the monadicity theorem (46) and the comparison statement (49) – as its classical counterpart in Prop. 4.5, relying on the fact that $\otimes \mathbb{1}_W$ is conservative (by the same argument as before) and both a left and a right adjoint. ☐

Remark 4.10 (Homomorphisms of free $\Diamond/\Box$-modales)**.** More explicitly,
(i) for some $G_\bullet : \Diamond_W \mathcal{H}_\bullet \to \Diamond_W \mathcal{K}_\bullet$ to be a homomorphism of (free) $\Diamond$-modales, it needs to make the following square commute:

$$\begin{tikzcd}
\underset{W}{\Diamond}\,\underset{W}{\Diamond}\,\mathcal{H}_\bullet \arrow[rrr, "\mathrm{join}^{\Diamond_W}_{\mathcal{H}_\bullet}"] \arrow[dddd, "\underset{W}{\Diamond} G_\bullet"'] & & & \underset{W}{\Diamond}\,\mathcal{H}_\bullet \arrow[dddd, "G_\bullet"] \\
 & \underset{w''}{\bigoplus} |\psi_{w,w',w''}\rangle \arrow[r, mapsto] \arrow[dd, mapsto] & \underset{w''}{\sum}\,\underset{w'}{\bigoplus}\, |\psi_{w,w',w''}\rangle \arrow[d, mapsto] & \\
 & & G_w \underset{w''}{\sum}\,\underset{w'}{\bigoplus}\, |\psi_{w,w',w''}\rangle & \\
 & \underset{w''}{\bigoplus} G_{w''} \underset{w'}{\bigoplus} |\psi_{w,w',w''}\rangle \arrow[r, mapsto] & \underset{w''}{\sum} G_{w''} \underset{w'}{\bigoplus} |\psi_{w,w',w''}\rangle \arrow[u, equals] & \\
\underset{W}{\Diamond}\,\underset{W}{\Diamond}\,\mathcal{K}_\bullet \arrow[rrr, "\mathrm{join}^{\Diamond_W}_{\mathcal{K}_\bullet}"] & & & \underset{W}{\Diamond}\,\mathcal{H}_\bullet
\end{tikzcd}$$

This is clearly possible only if G_w is actually independent of w, ie. if $G_\bullet = G := G \otimes \mathbb{1}_W$.
(ii) Analogously for homomorphisms of free $\Box$-modales:

$$\begin{tikzcd}
\underset{W}{\Box}\mathcal{H}_\bullet \arrow[rrr, "\mathrm{dplc}^{\Box_W}_{\mathcal{H}_\bullet}"] \arrow[dddd, "G_\bullet"'] & & & \underset{WW}{\Box\Box}\mathcal{H}_\bullet \arrow[dddd, "\underset{W}{\Box} G_\bullet"] \\
 & \underset{w'}{\bigoplus} |\psi_{w,w'}\rangle \arrow[r, mapsto] \arrow[dd, mapsto] & \underset{w''}{\bigoplus}\,\underset{w'}{\bigoplus} |\psi_{w,w'}\rangle \arrow[d, mapsto] & \\
 & & \underset{w''}{\bigoplus} G_{w''} \underset{w'}{\bigoplus} |\psi_{w,w'}\rangle & \\
 & G_w \underset{w'}{\bigoplus} |\psi_{w,w'}\rangle \arrow[r, mapsto] & \underset{w''}{\bigoplus} G_w \underset{w'}{\bigoplus} |\psi_{w,w'}\rangle \arrow[u, equals] & \\
\underset{W}{\Box}\mathcal{K}_\bullet \arrow[rrr, "\mathrm{dplc}^{\Box_W}_{\mathcal{K}_\bullet}"] & & & \underset{WW}{\Box\Box}\mathcal{K}_\bullet
\end{tikzcd}$$

In summary so far, we have found a quantum epistemic logic with the following interpretations, analogous to (138):

$$\begin{tikzcd}[column sep=small]
 & & & & \text{principle of quantum compulsion:} & & \\
\textbf{necessarily } \mathcal{H}_\bullet & \textbf{entails} & \textbf{actually } \mathcal{H}_\bullet & \textbf{entails} & \textbf{possibly } \mathcal{H}_\bullet & \textbf{is} & \textbf{necessarily } \mathcal{H}_\bullet \\
\Box_W \mathcal{H}_\bullet \arrow[rr, "\mathrm{obt}^{\Box_W}_{\mathcal{H}_\bullet}" description] & & \mathcal{H}_\bullet \arrow[rr, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}" description] & & \Diamond_W \mathcal{H}_\bullet & \simeq & \Box_W \mathcal{H}_\bullet
\end{tikzcd}$$

$$\begin{tikzcd}
\textbf{In world} & \textbf{observe...} & & & & \\
w : W & \vdash & \mathcal{H} \arrow[r, two heads, "\oplus_{w'} |\psi_{w'}\rangle \;\mapsto\; |\psi_w\rangle", "\textbf{measurement collapse}"'] \arrow[rr, bend right, "\textbf{linear projector onto sub-Hilbert space } \mathcal{H}_w"'] & \mathcal{H}_w \arrow[r, hook, "|\psi_w\rangle \mapsto \oplus_{w'} \delta_w^{w'} |\psi_w\rangle", "\textbf{state preparation}"'] & \mathcal{H}, & \text{where } \mathcal{H} := \bigoplus_{w':W} \mathcal{H}_{w'}
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
\textbf{randomly } \mathcal{H} & \textbf{entails} & \textbf{potentially } \mathcal{H} & \textbf{entails} & \textbf{indefinitely } \mathcal{H} \\
\text{☆}_W \mathcal{H} \arrow[rr, "\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}}" description] & & \mathcal{H} \arrow[rr, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}" description] & & \bigcirc_W \mathcal{H} \\
\bigoplus_{w:W} \mathcal{H} \arrow[rr, "\oplus_W |\psi_W\rangle \mapsto \sum_W |\psi_W\rangle", "\textbf{quantum superposition}"'] & & \mathcal{H} \arrow[rr, "|\psi\rangle \mapsto \oplus_{w':W} |\psi\rangle", "\textbf{quantum parallelism}"'] & & \bigoplus_{b:B} \mathcal{H}
\end{tikzcd}$$

(142)

However, for linear types, we have yet another monadicity statement:

Proposition 4.11 (Actual quantum data as indefiniteness-modal potential data)**.** *For* $W : \mathrm{FinClaType}_\Gamma$ *and* $p_W : W \to \Gamma$ *an epimorphism, the dependent sum* $\oplus_W : \mathrm{QuType}_W \to \mathrm{QuType}_\Gamma$ *is also monadic, whence the actual quantum types are identified with the (free) randomness/infiniteness modal types:*

$$\begin{tikzcd}
 & & & \textbf{randomly} & \\
\textbf{Randomness modal data} & \mathrm{QuType}_\Gamma^{\text{☆}_W} \arrow[d, phantom, "\wr|"] \arrow[r, "\oplus_W"] & {} \arrow[d, phantom, "\perp"] & & \\
\textbf{Actual quantum data} & \mathrm{QuType}_W \arrow[d, phantom, "\wr|"] & {} \arrow[d, phantom, "\perp"] & \mathrm{QuType}_\Gamma \arrow[ll, "\otimes \mathbb{1}_W" description] \arrow[loop above, "\text{☆}_W"] \arrow[loop below, "\bigcirc_W"] \arrow[r, phantom, "\perp"] & \textbf{Potential quantum data} \\
\textbf{Indefiniteness modal data} & \mathrm{QuType}_\Gamma^{\bigcirc_W} \arrow[r, "\oplus_W"'] & {} & & \\
 & & & \textbf{indefinitely} &
\end{tikzcd}$$

(143)

Proof. Due to ambidexterity (140) for finite W, in the quantum case also $\oplus_W$ is both a left and right adjoint, as shown. Therefore the monadicity theorem (46) implies the claim for $\bigcirc_W$ by observing that $\oplus_W$ is conservative. This is indeed the case, as it sends a morphism to its world-wise application, which is an isomorphism of dependent types if and only if it is so world-wise, hence if and only the original morphisms was so. The dual claim for the adjoint comonad ☆ now follows by (49). $\Box$

Remark 4.12 (Effective perspective on quantum epistemology)**.** Prop. 4.11 says that (over a finite inhabited type of classical worlds W) dependent linear types are $\bigcirc$-monadic! But since we have seen that dependent linear types may be thought of as quantum states in "many worlds", this gives a monadic perspective on quantum epistemology which allows for speaking about it in terms of *computational effects* (Lit. 2.15).

Hence we shall refer to these equivalent perspectives as the *epistemic* and the *effective* perspective, respectively:

$$\begin{tikzcd}
\textbf{Epistemic perspective} \arrow[ddd, leftrightarrow, "\text{monadicity}", "\text{of } \oplus_W"'] & \mathrm{QuType}_W \arrow[ddd, shift left=2, "\bigoplus_W"] \arrow[ddd, phantom, "\dashv"] & \mathcal{H}_\bullet \arrow[r, "G_\bullet" description] & \mathcal{K}_\bullet & \mathcal{H} & \mathcal{H} \xrightarrow{\;G_\bullet\;} \mathcal{H} \\
 & & \textbf{map of } W\textbf{-dependent types} \arrow[d, leftrightarrow] & & \textbf{in-dependent type} \arrow[d, leftrightarrow] & W\textbf{-dependent map of in-dependent types} \arrow[d, leftrightarrow] \\
 & & \textbf{homomorphism of } \bigcirc_W\textbf{-modales} & & \textbf{free } \bigcirc_W\textbf{-modale} & \bigcirc_W\textbf{-Kleisli map} \\
\textbf{Effective perspective} & \mathrm{QuType}_\Gamma \arrow[uuu, shift left=2, "\otimes \mathbb{1}_W"] \arrow[loop below, "\underset{W}{\bigcirc}"] & \bigoplus_W \mathcal{H}_\bullet \arrow[r, "\bigoplus_W G_\bullet" description] \arrow[loop below, "\underset{W}{\bigcirc}"] & \bigoplus_W \mathcal{K}_\bullet \arrow[loop below, "\underset{W}{\bigcirc}"] & \underset{W}{\bigcirc} \mathcal{H} & \mathrm{bind}\Big(\mathcal{H} \xrightarrow[\bigoplus_W G_\bullet \circ \mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}]{} \underset{W}{\bigcirc} \mathcal{K}\Big)
\end{tikzcd}$$

(144)

The effective perspective on the epistemic entailments (142) yields an effect-language for quantum measurement and controlled quantum gates – this we discuss next in §4.3.

Remark 4.13 (Relation to `zxCalculus`)**.** Something close to the identification $\big(\mathrm{QuType}_{\mathrm{r}}\big)^{\text{☆}_W} \simeq \mathrm{QuType}_W$ (in Prop. 4.11) has previously been observed in [CPav08, Thm. 1.5] (cf. Lit. 2.16), subject to some translation which we discuss now.

Frobenius-algebraic formulation. Remarkably, the above modal quantum logic gives rise to the "classical-structures" Frobenius monads used in the `zxCalculus` (Lit. 2.16). In particular this shows that/how `LHoTT/QS` can be used for certifying (type-checking) `zxCalculus`-protocols:

Proposition 4.14 (**Quantum (co)effects via Frobenius algebra).**

(i) *For W : ClaType, the W-(co)reader (co)monad on linear types (§4.2) is equivalent to the linear version $QW \otimes (\text{-})$ of the (co)writer (co)monad* (33) *induced by the canonical (co)algebra structure on $QW \equiv \bigoplus_W \mathbb{1}$;*

(ii) *If W : FinClaType is finite then the underlying functors of all these (co)monads agree and make a single Frobenius monad induced from the canonical Frobenius-algebra structure on $QW = \underset{W}{\bigoplus}\mathbb{1}$ (cf. Lit. 2.16):*

Frobenius structure on $QW = \bigoplus_W \mathbb{1}$	
algebra structure	**coalgebra structure**
$\mathbb{1} \xrightarrow{\mathrm{unit}_{QW}} QW$ $1 \mapsto \oplus_w \lvert w\rangle$	$QW \xrightarrow{\mathrm{counit}_{QW}} \mathbb{1}$ $\lvert w\rangle \mapsto 1$
$QW \otimes QW \xrightarrow{\mathrm{prod}_{QW}} QW$ $\lvert w_1\rangle \otimes \lvert w_2\rangle \mapsto \delta^{w_2}_{w_1}\lvert w_2\rangle$	$QW \xrightarrow{\mathrm{coprod}_{QW}} QW \otimes QW$ $\lvert w\rangle \mapsto \lvert w\rangle \otimes \lvert w\rangle$

quantum indefiniteness				quantum randomness
quantum reader		quantum (co)writer		quantum co-reader
$\underset{W}{\bigcirc}$	$\simeq$	$QW \otimes (\text{-})$	$\simeq$	$\underset{W}{\text{☆}}$
Monads	$\leftarrow$	FrobMonads	$\rightarrow$	CoMonads

Proof. With Prop. 4.8, this is a straightforward matter of unwinding the definitions:

$$\begin{tikzcd}
\textbf{measurement} & & & \\
\textbf{epistemic} & \underset{W}{\Box}\big(\underset{W}{\mathbb{1}}\otimes\mathcal{H}\big) \arrow[r, "\mathrm{obt}^{\Box_W}_{\mathcal{H}\otimes\mathbb{1}_W}"] \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] & \underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] & \\
\textbf{effective} & \underset{W}{\bigcirc}\,\underset{W}{\bigcirc}\,\mathcal{H} \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}}"] \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\bigcirc}\,\mathcal{H} \arrow[d, leftrightarrow, "\wr"'] & \mathcal{H} \arrow[l, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}"'] \arrow[d, leftrightarrow, "\wr"'] \\
\textbf{algebraic} & QW\otimes QW\otimes\mathcal{H} \arrow[r, "\mathrm{prod}_{QW}\otimes\mathrm{id}_{\mathcal{H}}", "\lvert w_1, w_2\rangle\otimes\lvert\psi\rangle \mapsto \delta^{w_2}_{w_1}\lvert w_2\rangle\otimes\lvert\psi\rangle"'] & QW\otimes\mathcal{H} & \mathbb{1}\otimes\mathcal{H} \arrow[l, "\mathrm{unit}_{QW}\otimes\mathrm{id}_{\mathcal{H}}"', "\sum_w \lvert w\rangle\otimes\lvert\psi\rangle \leftarrow\!\shortmid 1\otimes\lvert\psi\rangle"] \\
\textbf{state preparation} & & & \\
\textbf{epistemic} & & \underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[r, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}\otimes\mathbb{1}_W}"] \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] & \underset{W}{\Diamond}\,\underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] \\
\textbf{effective} & \mathcal{H} \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\text{☆}}\mathcal{H}T \arrow[l, "\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}}"'] \arrow[r, "\mathrm{dplc}^{\text{☆}_W}_{\mathcal{H}}"] \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\text{☆}}\,\underset{W}{\text{☆}}\mathcal{H} \arrow[d, leftrightarrow, "\wr"'] \\
\textbf{algebraic} & \mathcal{H} & QW\otimes\mathcal{H} \arrow[l, "\mathrm{counit}_{QW}"', "\lvert\psi\rangle \leftarrow\!\shortmid \lvert w\rangle\otimes\lvert\psi\rangle"] \arrow[r, "\mathrm{coprod}_{QW}\otimes\mathrm{id}_{\mathcal{H}}", "\lvert w\rangle\otimes\lvert\psi\rangle \mapsto \lvert w\rangle\otimes\lvert w\rangle\otimes\lvert\psi\rangle"'] & QW\otimes QW\otimes\mathcal{H}
\end{tikzcd} \tag{145}$$

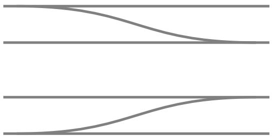
$\square$

In fact, this Frobenius structure is "*special*" in that

$$\begin{tikzcd}
\underset{W}{\text{☆}} \arrow[r, "\mathrm{dplc}^{\text{☆}_W}"] \arrow[rrr, bend right, "\sim"'] & \underset{W}{\text{☆}}\,\underset{W}{\text{☆}} \arrow[r, phantom, "\simeq"] & \underset{W}{\bigcirc}\,\underset{W}{\bigcirc} \arrow[r, "\mathrm{join}^{\bigcirc_W}"] & \underset{W}{\bigcirc}
\end{tikzcd} \tag{146}$$

Remark 4.15 (Frobenius property and Spider theorem)**.** The Frobenius property of $\bigcirc \simeq \text{☆}$ (Prop. 4.14) says explicitly that this diagram commutes:

$$\begin{tikzcd}
& \underset{W}{\text{☆}}\underset{W}{\text{☆}}\underset{W}{\bigcirc} \arrow[r, phantom, "\simeq"] & \underset{W}{\text{☆}}\underset{W}{\bigcirc}\underset{W}{\bigcirc} \arrow[dr, "{}^{\text{☆}}_{W}\mathrm{join}^{\bigcirc_W}_{(\text{-})}"] & \\
\underset{W}{\text{☆}}\underset{W}{\bigcirc} \arrow[ur, "\mathrm{dplc}^{\text{☆}_W}_{\bigcirc_W(\text{-})}"] \arrow[d, phantom, "\wr|"] & & & \underset{W}{\text{☆}}\underset{W}{\bigcirc} \arrow[d, phantom, "|\wr"] \\
\underset{W}{\bigcirc}\underset{W}{\text{☆}} \arrow[dr, "{}^{\bigcirc}_{W}\mathrm{dplc}^{\text{☆}_W}_{(\text{-})}"'] & & & \underset{W}{\bigcirc}\underset{W}{\text{☆}} \\
& \underset{W}{\bigcirc}\underset{W}{\text{☆}}\underset{W}{\text{☆}} \arrow[r, phantom, "\simeq"] & \underset{W}{\bigcirc}\underset{W}{\bigcirc}\underset{W}{\text{☆}} \arrow[ur, "\mathrm{join}^{\bigcirc_W}_{\text{☆}_W(\text{-})}"'] &
\end{tikzcd}$$



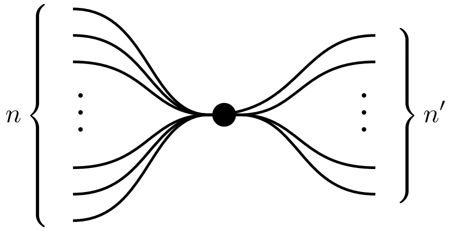
but this already implies (by the theory of *normal forms* [Ab96, Prop. 12, Fig. 3][Ko04], together with specialty (146))the equality of all those transformations of the form

$$\bigcirc^{n} \longrightarrow \text{☆}^{n'} \tag{147}$$

which arise as composites of $\bigcirc$-joins and of ☆-duplicates and which are *connected* in that there is no non-trivial horizontal decomposition such as in this simple disconnected example:

$$\begin{array}{c}
\underset{W}{\bigcirc}\,\underset{W}{\bigcirc}\,\underset{W}{\bigcirc}\,\mathcal{H} \xrightarrow{\qquad \mathrm{join}^{\bigcirc_W}_{\bigcirc_W \mathcal{H}} \qquad} \underset{W}{\bigcirc}\,\underset{W}{\bigcirc}\,\mathcal{H} \xrightarrow{\quad \mathrm{join}^{\bigcirc_W}_{\mathcal{H}} \quad} \underset{W}{\bigcirc}\,\mathcal{H} \simeq \underset{W}{\text{☆}}\,\mathcal{H} \xrightarrow{\quad \mathrm{dplc}^{\text{☆}_W}_{\mathcal{H}} \quad} \underset{W}{\text{☆}}\,\underset{W}{\text{☆}}\,\mathcal{H} \\[1em]
\mathrm{Q}W \otimes \mathrm{Q}W \otimes \mathrm{Q}W \otimes \mathcal{H} \xrightarrow{\mathrm{prod}_{\mathrm{Q}W} \otimes \mathrm{id}_{\mathrm{Q}W}} \mathrm{Q}W \otimes \mathrm{Q}W \otimes \mathcal{H} \xrightarrow{\mathrm{prod}_{\mathrm{Q}W} \otimes \mathrm{id}_{\mathcal{H}}} \mathrm{Q}W \otimes \mathcal{H} \xrightarrow{\mathrm{coprod}_{\mathrm{Q}W} \otimes \mathrm{id}_{\mathcal{H}}} \mathrm{Q}W \otimes \mathrm{Q}W \otimes \mathcal{H}
\end{array}$$

This classical fact of Frobenius algebra theory has been called the *spider theorem* in [CD08, Thm. 1], since it means that in string diagram notation, all the operations (147) may uniquely by depicted by a diagram of this form:



(148)

These are the *spider diagrams* used in `zxCalculus` (Lit. 2.16).

4.3 Controlled quantum gates

We explain how *controlled quantum gates* and *quantum measurement gates* (Lit. 2.1) are naturally represented in the quantum modal logic of §4.2 and give (Prop. 4.16) a formal proof of the *deferred measurement principle* (9).

Data-typing of controlled quantum gates via quantum modal types.

We may observe that, with §4.2, we now have available the natural data-typing of classical/quantum data that is indicated on the right.

Notice how the distinction between classical and quantum data is reflected by the application or not of the (co)monad $\bigcirc$ ($\Box$).

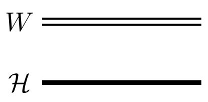
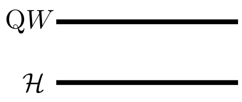
Throughout we use monadicity of $\bigoplus_W$ (Prop. 4.11) to translate (144)

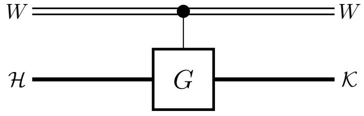
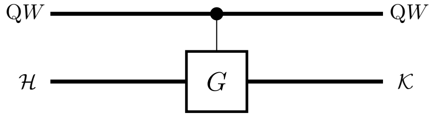
- *epsistemic typing*
  via W-dependent linear types

into

- *effective typing*
  via $\bigcirc_W$-modal linear types.

Besides the practical utility which we demonstrate in the following, the modal logic of this typing neatly reflects intuition, as shown.

	Classical/quantum register	**Controlled quantum register**
Symbolic		
Epistemic	actual quantum data $\dfrac{\mathcal{H}_\bullet \;:\; \mathrm{QuType}_W}{w : W \;\vdash\; \mathcal{H}_w \;:\; \mathrm{QuType}}$	potential quantum data $\dfrac{\underset{W}{\Box}\mathcal{H}_\bullet \;:\; \mathrm{QuType}_W}{w : W \;\vdash\; \underset{w'}{\bigoplus}\mathcal{H}_{w'} \;:\; \mathrm{QuType}}$
Effective	indefiniteness-handling quantum data $$\begin{tikzcd} \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] & : & \mathrm{QuType}^{\bigcirc_W} \end{tikzcd}$$	free indefiniteness-handling quantum data $$\begin{tikzcd} \underset{W}{\bigoplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] & : & \mathrm{QuType}^{\bigcirc_W} \\ \underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{H}_\bullet & : & \mathrm{QuType}_{\bigcirc_W} \arrow[u, hook] \end{tikzcd}$$

	Classically controlled quantum gate	**Quantumly controlled quantum gate**
Symbolic		
Epistemic	$$\begin{tikzcd} & \mathcal{H}_\bullet \arrow[r, "G_\bullet", "\text{an actual entailment}"'] & \mathcal{K}_\bullet \\ w : W \;\vdash & \mathcal{H}_w \arrow[r, "G_w"] & \mathcal{K}_w \end{tikzcd}$$	$$\begin{tikzcd} & \underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\Box}G_\bullet", "\text{a potential entailment}"'] & \underset{W}{\Box}\mathcal{K}_\bullet \\ w : W \;\vdash & \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\bigoplus}G_\bullet" description] & \underset{W}{\bigoplus}\mathcal{K}_\bullet \end{tikzcd}$$
Effective	$$\begin{tikzcd} \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\bigoplus}G_\bullet"] \arrow[d, equal, "\text{if } \mathcal{H}_\bullet = \mathcal{H}"'] & \underset{W}{\bigoplus}\mathcal{K}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal, "\text{if } \mathcal{K}_\bullet = \mathcal{K}"'] \\ \underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\underset{W}{\bigcirc}G_\bullet", "\mathrm{bind}\left(\mathcal{H} \xrightarrow{G_\bullet} \bigcirc_W \mathcal{K}\right)"'] & \underset{W}{\bigcirc}\mathcal{K} \end{tikzcd}$$ a $\bigcirc$-effectful operation	$$\begin{tikzcd} \underset{W}{\bigoplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\bigoplus}\underset{W}{\Box}G_\bullet"] \arrow[d, equal] & \underset{W}{\bigoplus}\underset{W}{\Box}\mathcal{K}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] \\ \underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\bigcirc}\underset{W}{\bigoplus}G_\bullet", "\mathrm{bind}\left(\mathrm{return} \circ \underset{W}{\bigoplus}G_\bullet\right)"'] & \underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{K}_\bullet \end{tikzcd}$$ a $\bigcirc$-effectless operation

Here the "epistemic"-typing of controlled quantum gates shown in the middle row is manifest: For classical control the quantum gate is a W-dependent linear map, while for quantum control it is a genuine linear map on the W-indexed direct sum. The equivalent (144) "effective" typing in the top line of the bottom row follows by monadicity of $\bigoplus_W$ (see Prop. 4.11). The very last line shows the corresponding Kleisli-triple formulation of "programs with side effects" (21). On the left this requires assuming that the dependent linear type is constant, $\mathcal{H}_\bullet = \mathcal{H}$ (which typically is the case in practice, see the example on p. 68) since that makes it correspond to a free $\bigcirc$-modale. On the right we see the effectless operation (22).

Quantum measurement – Copenhagen-style. Last but not least, we obtain this way a natural typing of the otherwise subtle case of quantum measurement gates: These are now given simply by the $\Box$-counit and, equivalently, by the $\bigcirc$-join (cf. Prop. 4.8), as shown on the right.

Via the language of effectful computation (Lit. 2.15) and with the "reader-monad" $\bigcirc$ modally pronounced as "indefiniteness" (136), this translates to the pleasant statement that:

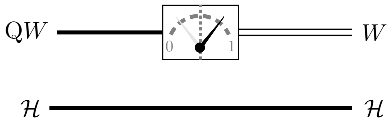
"For effectively-typed quantum data, quantum measurement is nothing but the *handling of indefiniteness-effects.*"

In more detail:
"Before measurement, quantum data is indefinite(-effectful), while quantum measurement actualizes the data by handling of its indefiniteness(-effect)"

This way the puzzlement of the "state collapse" (12) is resolved into an appropriate quantum effect language equivalent (144) to quantum modal logic.

Quantum measurement gate

Symbolic



Epistemic

$$\begin{tikzcd}
 & \underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\mathrm{obt}^{\Box_W}_{\mathcal{H}_\bullet}", "\text{the necessary becomes actual}"'] & \mathcal{H}_\bullet \\
w : W \vdash & \underset{w'}{\bigoplus}\mathcal{H}_{w'} \arrow[r, "\mathrm{pr}_w", "\text{quantum state collapse}"'] & \mathcal{H}_w \\
 & \underset{w'}{\bigoplus}|\psi_{w'}\rangle \arrow[r, mapsto] & |\psi_w\rangle
\end{tikzcd}$$

Effective

$$\begin{tikzcd}
\underset{W}{\bigoplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\bigoplus}\mathrm{obt}^{\Box_W}_{\mathcal{H}_\bullet}"] \arrow[d, equal] \arrow[dr, phantom, "(44)"] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] \\
\underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathrm{hndl}^{\bigcirc_W}_{\bigoplus_W \mathcal{H}_\bullet}", "\bigcirc_W\text{-effect handling}"'] & \underset{W}{\bigoplus}\mathcal{H}_\bullet
\end{tikzcd}$$

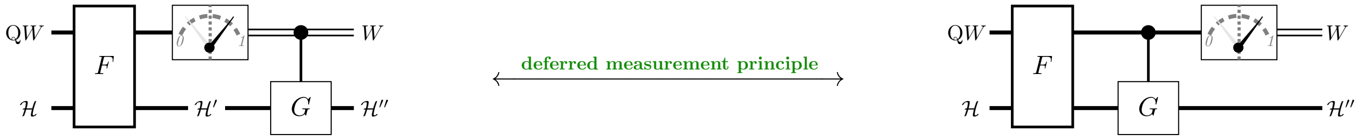
Before looking at examples (p. 68), we record a basic structural result immediately implied by this typing, which may evidently be understood as formalizing the *deferred measurement principle* (9), thus making this principle verifiable in `LHoTT` as [Sta15] envisioned should be the case for any respectable quantum programming language:

Proposition 4.16 (Deferred measurement principle). *With respect to the above typing of quantum gates, the* $\Box$*-Kleisli equivalence* (42) *is the following transformation of quantum circuits:*

$$\begin{tikzcd}
(\mathrm{QuType}_W)_{\Box_W}(\mathcal{H}_\bullet, \mathcal{H}''_\bullet) \arrow[r, "\sim", "\underset{W}{\Box}(\text{-}) \circ \mathrm{dplc}^{\Box_W}_{(\text{-})}"'] \arrow[rr, bend left, "\mathrm{id}", "\text{Kleisli equivalence}"'] & \mathrm{QuType}^{\Box_W}_W\left(\underset{W}{\Box}\mathcal{H}_\bullet, \underset{W}{\Box}\mathcal{H}''_\bullet\right) \arrow[r, "\sim", "\mathrm{obt}^{\Box_W}_{(\text{-})} \circ (-)"'] & (\mathrm{QuType}_W)_{\Box_W}(\mathcal{H}_\bullet, \mathcal{H}''_\bullet) \\
\underset{W}{\Box}\text{-Kleisli morphisms} & \text{homomorphisms of free } \underset{W}{\Box}\text{-coalgebras} & \underset{W}{\Box}\text{-Kleisli morphisms}
\end{tikzcd}$$

$$\left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\mathrm{obt}^{\Box_W}_{\mathcal{H}'_\bullet}} \mathcal{H}'_\bullet \xrightarrow{G_\bullet} \mathcal{H}''_\bullet\right) \mapsto \left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\Box_W G_\bullet} \underset{W}{\Box}\mathcal{H}''_\bullet\right) \mapsto \left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\Box_W G_\bullet} \underset{W}{\Box}\mathcal{H}''_\bullet \xrightarrow{\mathrm{obt}^{\Box_W}_{\mathcal{H}''_\bullet}} \mathcal{H}''_\bullet\right)$$

measurement-controlled quantum gate · quantum-controlled quantum gate... · ...followed by measurement



Proof. It just remains to see that the Kleisli equivalence $\underset{W}{\Box}(\text{-}) \circ \mathrm{dplc}^{\Box_W}_{(\text{-})}$ acts in the first step as claimed, hence that the following diagram commutes:

$$\begin{tikzcd}
\underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "F"] \arrow[d, "\mathrm{dplc}^{\Box_W}_{\mathcal{H}_\bullet}"'] & \underset{W}{\Box}\mathcal{H}'_\bullet \arrow[d, "\mathrm{dplc}^{\Box_W}_{\mathcal{H}'_\bullet}"'] \arrow[dr, equal] & & \\
\underset{W}{\Box}\,\underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\Box_W F"'] & \underset{W}{\Box}\,\underset{W}{\Box}\mathcal{H}'_\bullet \arrow[r, "\underset{W}{\Box}\left(\mathrm{obt}^{\Box_W}_{\mathcal{H}'_\bullet}\right)"'] & \underset{W}{\Box}\mathcal{H}'_\bullet \arrow[r, "\Box_W G_\bullet"'] & \underset{W}{\Box}\mathcal{H}''_\bullet
\end{tikzcd}$$

But the square commutes since the gate F is independent of the measurement result $w : W$ and hence is a homomorphism of free $\Box$-coalgebras (by Rem. 4.10), while the triangle commutes by the comonad axioms (24). $\Box$

Example: Modal typing of basic QBit-gates. The key aspects of the above modal typing rules for quantum gates are already well-illustrated by simple examples of standard QBit-gates such as the CNOT-gate (8).

Here the quantum state space is that of a pair of coupled qbits, $\mathrm{QBit} \otimes \mathrm{QBit}$, and the "many possible worlds" $W \equiv \mathrm{Bit}$ are labeled by the bits which are the classical outcomes of measurements on the first qbit in the pair:

$$
\begin{aligned}
\mathrm{Bit} &\equiv \{0,\,1\} && \in \mathrm{ClaType}\,,\\
\mathrm{QBit} &\equiv \mathbb{C}\big[\{0,\,1\}\big] \simeq \mathbb{C}^2 && \in \mathrm{QuType}\,.
\end{aligned}
$$

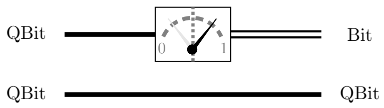
In seeing how the modal typing shown on the right and below matches the standard formulas (8) we repeatedly make use of the following canonical identifications:

$$
\begin{aligned}
& \mathrm{QBit} \otimes \mathrm{QBit}\\
\simeq\;& \mathbb{C}[\mathrm{Bit}] \otimes \mathrm{QBit}\\
\simeq\;& \big(\mathbb{C}_0 \oplus \mathbb{C}_1\big) \otimes \mathrm{QBit}\\
\simeq\;& \mathrm{QBit}_0 \oplus \mathrm{QBit}_1\\
\simeq\;& \textstyle\bigoplus_{\mathrm{Bit}} \mathrm{QBit}_\bullet\\
\simeq\;& \bigcirc_{\mathrm{Bit}} \mathrm{QBit}\;,
\end{aligned}
$$

where the subscript indicates which direct summand corresponds to which "branch" of "worlds" of possible measurement outcomes.

QBit-Measurement

symbolic



epistemic

$$\begin{tikzcd} \underset{\mathrm{Bit}}{\Box}\,\mathrm{QBit}_\bullet \arrow[r, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathrm{QBit}_\bullet}"] & \mathrm{QBit}_\bullet \end{tikzcd}$$

$$\begin{tikzcd} b : \mathrm{Bit} \;\vdash & \mathrm{QBit} \otimes \mathrm{QBit} \arrow[r] & \mathrm{QBit} \\ & \lvert b_1 \rangle \otimes \lvert b_2 \rangle \arrow[r, phantom, "\mapsto"] & \delta^{b_1}_{b} \lvert b_2 \rangle \end{tikzcd}$$

Effective

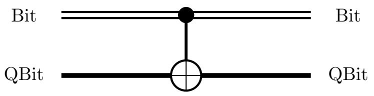
$$\begin{tikzcd} \underset{\mathrm{Bit}}{\bigcirc}\;\underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \arrow[r, "\mathrm{hndl}^{\bigcirc_{\mathrm{Bit}}}_{\bigcirc_{\mathrm{Bit}}\mathrm{QBit}}"] & \underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \end{tikzcd}$$

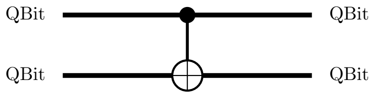
$$\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[dr, "P_0\otimes\mathrm{id}"] & \\ \oplus & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "P_1\otimes\mathrm{id}"'] & \end{tikzcd}$$

$$\begin{tikzcd} & \mathrm{QBit} \\ \mathrm{bind}\Big(\,\mathrm{QBit} \otimes \mathrm{QBit} \arrow[ur, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \delta^{b_1}_{0}\lvert b_2\rangle"] \arrow[dr, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \delta^{b_1}_{1}\lvert b_2\rangle"'] & \oplus\;\Big) \\ & \mathrm{QBit} \end{tikzcd}$$

CNOT gate

Symbolic





Epistemic

$$\begin{tikzcd} \mathrm{QBit}_\bullet \arrow[r, "\mathrm{CNOT}_\bullet"] & \mathrm{QBit}_\bullet \end{tikzcd}$$

$$\begin{tikzcd} b : \mathrm{Bit} \;\vdash & \mathrm{QBit} \arrow[r] & \mathrm{QBit} \\ & \lvert b_2 \rangle \arrow[r, phantom, "\mapsto"] & \lvert b \,\mathrm{xor}\, b_2 \rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{\mathrm{Bit}}{\Box}\,\mathrm{QBit}_\bullet \arrow[r, "\underset{\mathrm{Bit}}{\Box}\,\mathrm{CNOT}_\bullet"] & \underset{\mathrm{Bit}}{\Box}\,\mathrm{QBit}_\bullet \end{tikzcd}$$

$$\begin{tikzcd} b : \mathrm{Bit} \;\vdash & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r] & \mathrm{QBit}\otimes\mathrm{QBit} \\ & \lvert b_1 \rangle \otimes \lvert b_2 \rangle \arrow[r, phantom, "\mapsto"] & \lvert b_1 \rangle \otimes \lvert b_1 \,\mathrm{xor}\, b_2 \rangle \end{tikzcd}$$

Effective

$$\begin{tikzcd} \underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \arrow[r, "\underset{\mathrm{Bit}}{\bigoplus}\,\mathrm{CNOT}_\bullet"] & \underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \end{tikzcd}$$

$$\begin{tikzcd} \mathrm{QBit} \arrow[r, "\lvert b_2\rangle \mapsto \lvert 0 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit} \\ \oplus \arrow[r, phantom, "\oplus"] & \oplus \\ \mathrm{QBit} \arrow[r, "\lvert b_2\rangle \mapsto \lvert 1 \,\mathrm{xor}\, b_2\rangle"'] & \mathrm{QBit} \end{tikzcd}$$

$$\begin{tikzcd} & \mathrm{QBit} \\ \mathrm{bind}\Big(\,\mathrm{QBit} \arrow[ur, "\lvert b_2\rangle \mapsto \lvert 0 \,\mathrm{xor}\, b_2\rangle"] \arrow[dr, "\lvert b_2\rangle \mapsto \lvert 1 \,\mathrm{xor}\, b_2\rangle"'] & \oplus\;\Big) \\ & \mathrm{QBit} \end{tikzcd}$$

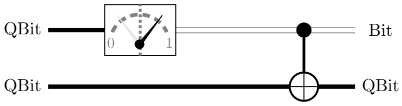
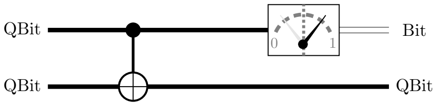
$$\begin{tikzcd} \underset{\mathrm{Bit}}{\bigcirc}\;\underset{\mathrm{Bit}}{\bigoplus}\,\mathrm{QBit} \arrow[r, "\underset{\mathrm{Bit}}{\bigcirc}\;\underset{\mathrm{Bit}}{\bigoplus}\,\mathrm{CNOT}_\bullet"] & \underset{\mathrm{Bit}}{\bigcirc}\;\underset{\mathrm{Bit}}{\bigoplus}\,\mathrm{QBit} \end{tikzcd}$$

$$\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\lvert b_1, b_2\rangle \mapsto \lvert b_1, b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \\ \oplus \arrow[r, phantom, "\oplus"] & \oplus \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\lvert b_1, b_2\rangle \mapsto \lvert b_1, b_1 \,\mathrm{xor}\, b_2\rangle"'] & \mathrm{QBit}\otimes\mathrm{QBit} \end{tikzcd}$$

$$\begin{tikzcd} & & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{bind}\Big(\,\mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\lvert b_1, b_2\rangle \mapsto \lvert b_1, b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur] \arrow[dr] & \oplus\;\Big) \\ & & \mathrm{QBit}\otimes\mathrm{QBit} \end{tikzcd}$$

For the record, we also spell out the two possible combinations of the above CNOT- and QBit-measurement gates:

CNOT with QBit-Measurement

<table>
<tr><td>symbolic</td><td> </td><td> </td></tr>
<tr><td>epistemic</td><td>

measurement **cls. contr. qnt. NOT**

$$\begin{tikzcd} \underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_{\bullet} \arrow[r, "\mathrm{obt}^{\Box_W}_{\mathrm{QBit}_\bullet}"] & \mathrm{QBit}_{\bullet} \arrow[r, "\mathrm{CNOT}_{\bullet}"] & \mathrm{QBit}_{\bullet} \end{tikzcd}$$

$$\begin{array}{lllllll} b : \mathrm{Bit} \;\vdash & \mathrm{QBit}\otimes\mathrm{QBit} & \longrightarrow & \mathrm{QBit}_b & \longrightarrow & \mathrm{QBit}_b \\ & \vert b_1\rangle \otimes \vert b_2\rangle & \mapsto & \vert b_2\rangle & \mapsto & \vert b \,\mathrm{xor}\, b_2\rangle \end{array}$$

</td><td>

quantum CNOT **measurement**

$$\begin{tikzcd} \underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_{\bullet} \arrow[r, "\underset{\mathrm{Bit}}{\Box}\mathrm{CNOT}_{\bullet}"] & \underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_{\bullet} \arrow[r, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathrm{QBit}_\bullet}"] & \mathrm{QBit}_{\bullet} \end{tikzcd}$$

$$\begin{array}{lllllll} b : \mathrm{Bit} \;\vdash & \mathrm{QBit}\otimes\mathrm{QBit} & \longrightarrow & \mathrm{QBit}\otimes\mathrm{QBit} & \longrightarrow & \mathrm{QBit}_b \\ & \vert b_1\rangle \otimes \vert b_2\rangle & \mapsto & \vert b_1\rangle \otimes \vert b_2 \,\mathrm{xor}\, b_1\rangle & \mapsto & \vert b_2 \,\mathrm{xor}\, b\rangle \end{array}$$

</td></tr>
<tr><td>Effective</td><td>

$$\underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit} \xrightarrow{\mathrm{hndl}^{\bigcirc_{\mathrm{Bit}}}_{\mathrm{QBit}}} \underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit} \xrightarrow{\underset{\mathrm{Bit}}{\bigoplus}\mathrm{CNOT}_{\bullet}} \underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit}$$

$$\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[dr, "P_{0}\otimes\mathrm{id}"] & & \\ \oplus & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\vert b_1\rangle\otimes\vert b_2\rangle \mapsto \vert b_1\rangle\otimes\vert b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "P_{1}\otimes\mathrm{id}"'] & & \end{tikzcd}$$

$$\mathrm{bind}\left(\begin{tikzcd} & \mathrm{QBit} \arrow[r, "\vert b_2\rangle\mapsto\vert 0 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "\vert b_1{,}b_2\rangle\mapsto\delta_{0}^{b_1}\,\vert b_2\rangle"] \arrow[dr, "\vert b_1{,}b_2\rangle\mapsto\delta_{1}^{b_1}\,\vert b_2\rangle"'] & \oplus & \oplus \\ & \mathrm{QBit} \arrow[r, "\vert b_2\rangle\mapsto\vert 1 \,\mathrm{xor}\, b_2\rangle"'] & \mathrm{QBit} \end{tikzcd}\right)$$

</td><td>

$$\underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit} \xrightarrow{\underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\bigoplus}\,\mathrm{QBit}} \underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit} \xrightarrow{\mathrm{hndl}^{\bigcirc}_{\bigoplus_{\mathrm{Bit}}\mathrm{QBit}}} \underset{\mathrm{Bit}}{\bigoplus}\mathrm{QBit}$$

$$\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\vert b_1\rangle\otimes\vert b_2\rangle \mapsto \vert b_1\rangle\otimes\vert b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[dr, "P_{0}\otimes\mathrm{id}"] & \\ \oplus & \oplus & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\vert b_1\rangle\otimes\vert b_2\rangle \mapsto \vert b_1\rangle\otimes\vert b_1 \,\mathrm{xor}\, b_2\rangle"'] & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "P_{1}\otimes\mathrm{id}"'] & \end{tikzcd}$$

$$\mathrm{bind}\left(\begin{tikzcd} & & \mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\vert b_1\rangle\otimes\vert b_2\rangle \mapsto \vert b_1\rangle\otimes\vert b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "\vert b_1{,}b_2\rangle\mapsto\delta_{0}^{b_1}\,\vert b_2\rangle"] \arrow[dr, "\vert b_1{,}b_2\rangle\mapsto\delta_{0}^{b_1}\,\vert b_2\rangle"'] & \oplus \\ & & \mathrm{QBit} \end{tikzcd}\right)$$

</td></tr>
</table>

Notice here how the component expressions on the left and right agree, in accord with the *deferred measurement principle* (Prop. 4.16). In components this is an elementary triviality, but the point is that by making this triviality follow from typing rules it becomes machine-verifiable also in more complex cases.

qRAM. As a byproduct of the modal typing of controlled quantum gates, we may notice a formal reflection of the idea of *circuit models for qRAM* (11). Namely if, with (36), we recall that RAM-effects are typed by the state monad $\underset{W}{\bigcirc}\underset{W}{\star}$ (137) — which immediately makes sense linearly just as it does classically—, then quantumly controlled quantum circuits in the above sense (p. 66) are formally identified with QRAM-effective quantum programs as follows, where the first transformation is for effectless programs (22) while the second is $\star_W \dashv \bigcirc_W$-adjointness (26):

The passage to circuit models for qRAM (11) may formally be understood as the modal adjointness between

(i) QRAM-effective quantum programs $\mathcal{H} \longmapsto \underset{W}{\bigcirc}\underset{W}{\star}\mathcal{K}$

(ii) quantumly controlled quantum circuits $\underset{W}{\bigoplus}\mathcal{H} \longmapsto \underset{W}{\bigoplus}\mathcal{K}$

$$\begin{tikzcd} \underset{W}{\bigcirc}\,\underset{W}{\bigoplus}\mathcal{H} \arrow[loop above, "\bigcirc_W"] \arrow[r, dash] & \underset{W}{\bigcirc}\,\underset{W}{\bigoplus}G_{\bullet} \arrow[r] & \underset{W}{\bigcirc}\,\underset{W}{\bigoplus}\mathcal{K} \arrow[loop above, "\bigcirc_W"] \end{tikzcd}$$

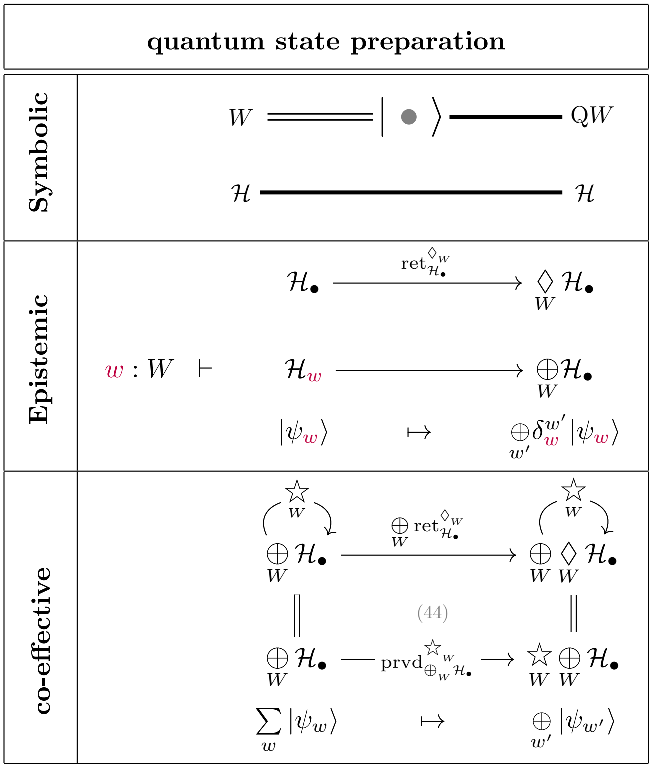
QW-controlled quantum gate (p. 66)

$$\begin{tikzcd} \underset{W}{\bigoplus}\mathcal{H} \arrow[r, dash] \arrow[d, equal] & \underset{W}{\bigoplus}G_{\bullet} \arrow[r] & \underset{W}{\bigoplus}\mathcal{K} \arrow[d, equal] \\ \underset{W}{\star}\mathcal{H} \arrow[r, dash] & \underset{W}{\bigoplus}G_{\bullet} \arrow[r] & \underset{W}{\star}\mathcal{K} \end{tikzcd} \tag{149}$$

$$\begin{tikzcd} \mathcal{H} \arrow[r, dash] & \widetilde{\underset{W}{\bigoplus}G_{\bullet}} \arrow[r] & \underset{W}{\bigcirc}\,\underset{W}{\star}\mathcal{K} \end{tikzcd}$$

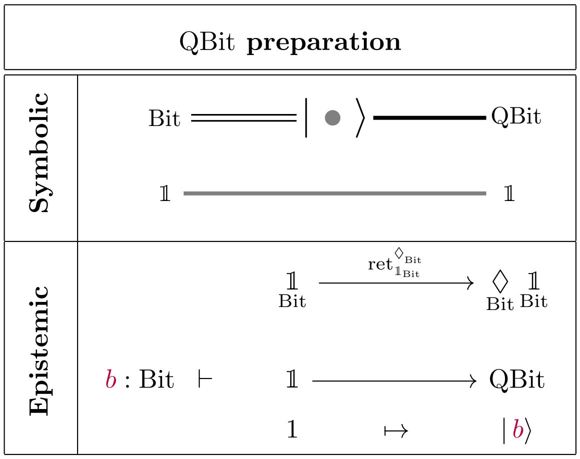
quantum circuit interacting with a QRAM space QW

Quantum contexts. The formal dual of the previous discussion of quantum measurement realized as a monadic computational effect yields *quantum state preparation* realized as a *comonadic computational context* (2.15): Shown on the left below is the modal typing of *quantum state preparation* in the generality of classical control, namely quantum state preparation conditioned on a classical parameter $w : W$. In the practice of quantum circuits, this typically applies to quantum types of the form $\underset{W}{\mathbb{1}}$ in which case the traditional notion of state preparation is manifest: In world w the result of the preparation is the quantum state $|w\rangle$. This is shown for the example of QBit-preparation on the right:



$$\begin{tikzcd}
 & \mathcal{H}_\bullet \arrow[r, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}"] & \underset{W}{\Diamond}\mathcal{H}_\bullet \\
w : W \ \vdash & \mathcal{H}_w \arrow[r] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \\
 & |\psi_w\rangle \arrow[r, mapsto] & \underset{w'}{\bigoplus}\delta^{w'}_{w}|\psi_w\rangle
\end{tikzcd}$$

$$\begin{tikzcd}
\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "\underset{W}{\star}"] \arrow[r, "\underset{W}{\bigoplus}\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}"] \arrow[d, equal] \arrow[dr, phantom, "(44)"] & \underset{W}{\bigoplus}\underset{W}{\Diamond}\mathcal{H}_\bullet \arrow[loop above, "\underset{W}{\star}"] \arrow[d, equal] \\
\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathrm{prvd}^{\star_W}_{\bigoplus_W\mathcal{H}_\bullet}"] & \underset{W}{\star}\underset{W}{\bigoplus}\mathcal{H}_\bullet \\
\sum\limits_{w}|\psi_w\rangle \arrow[r, mapsto] & \underset{w'}{\bigoplus}|\psi_{w'}\rangle
\end{tikzcd}$$



$$\begin{tikzcd}
 & \underset{\mathrm{Bit}}{\mathbb{1}} \arrow[r, "\mathrm{ret}^{\Diamond_{\mathrm{Bit}}}_{\mathbb{1}_{\mathrm{Bit}}}"] & \underset{\mathrm{Bit}}{\Diamond}\underset{\mathrm{Bit}}{\mathbb{1}} \\
b : \mathrm{Bit} \ \vdash & \mathbb{1} \arrow[r] & \mathrm{QBit} \\
 & 1 \arrow[r, mapsto] & |b\rangle
\end{tikzcd}$$

Quantum measurement – Everett style. But we may observe that quantum state preparation in the above classically-controlled generality can itself be used to model quantum measurement, namely as the *preparation of the collapsed state conditioned on the classical measurement outcome*!

This is seen from the last line of the co-effective typing above, which we recognize as the branching-perspective on quantum measurement – if only we disregard the $\star_W$-modale homomorphism property of this map – which formally corresponds to pulling this map back up by applying (-) $\otimes \mathbb{1}_W$. This yields the following purple map and hence the *Everett-style* typing of quantum measurement mentioned in the introduction (7) — which is related to the above Copenhagen-style typing (from p. 67) by the *hexagon of epistemic entailments* (4.2):

$$\begin{tikzcd}
 & \underset{W}{\square}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\square}G_\bullet"] & \underset{W}{\square}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\square}\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}", "\substack{\text{quantum measurement}\\ \text{typed Everett-style}}"'] & \underset{W}{\square}\underset{W}{\Diamond}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\square}\underset{W}{\Diamond}G_\bullet", "\mathrm{l}"'] & \underset{W}{\square}\underset{W}{\Diamond}\mathcal{H}_\bullet \\
w : W \ \vdash & \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\bigoplus}G_\bullet", "\text{coherent quantum gate}"'] \arrow[d, phantom, "|||"] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathrm{prvd}^{\star_W}_{\bigoplus_W\mathcal{H}_\bullet}", "\substack{\text{collapsed-state-preparation}\\ \text{by providing }\star\text{-context}}"'] \arrow[d, phantom, "|||"] & \underset{W}{\star}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "W"] \arrow[r, "\underset{W}{\star}\underset{W}{\bigoplus}G_\bullet", "\substack{\text{coherent quantum gate}\\ \text{under }\star}"'] \arrow[d, phantom, "|||"] & \underset{W}{\star}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "W"] \arrow[d, phantom, "|||"] \\
 & \mathcal{H} \arrow[r, "G"] & \mathcal{H} \arrow[r, "P_1"] \arrow[r, phantom, "\vdots\ \text{branching}"'] \arrow[r, "P_{|W|}"'] & \begin{matrix}\mathcal{H}\\ \oplus\\ \vdots\\ \oplus\\ \mathcal{H}\end{matrix} \arrow[r, "\begin{matrix}G\\ \oplus\\ \vdots\\ \oplus\\ G\end{matrix}"] & \begin{matrix}\mathcal{H}\\ \oplus\\ \vdots\\ \oplus\\ \mathcal{H}\end{matrix} \\
 & & \sum\limits_{w'}|\psi_{w'}\rangle \arrow[r, mapsto] & \underset{w''}{\bigoplus}|\psi_{w''}\rangle \arrow[r, mapsto] & \underset{w''}{\bigoplus}G_{w''}|\psi_{w''}\rangle
\end{tikzcd} \tag{150}$$

Remark 4.17 (No classical control appears in Everett-typing)**.** Comparing the epistemic hexagon (7), we find that where the Copenhagen-style typing sees a classically-controlled quantum gate (cf. p. 66) the Everett-style typing (150) sees (no classical control) but the corresponding quantumly-controlled quantum gate — but applied in each of several "branches".

This primacy of the non-classical quantum perspective and the disregard for the need of any classical contexts is what Everett amplified when speaking of the "universality" of the quantum state (this being the very title of his thesis [Ev57a]). The modal typing of quantum processes in (150) provides a formalization of this intuition in a precise and machine-verifiable form.

Remark 4.18 (Everett-style measurement typing in the literature)**.** Essentially the typing-by-branching of quantum measurement in the bottom of (150) may be recognized in the early proposal for quantum programming language syntax in [Se04, p. 568].

The observation (apparently independently of [Se04]) that this may usefully be understood as the prvd-operation of modales (coalgebras) over the comonad $\text{☆}_W \simeq \mathrm{Q}W \otimes (\text{-})$ (Prop. 4.14) is due to [CPav08, Thm. 1.5] (cf. [CPP0909, pp. 28]) — this being the origin of the Frobenius-monadic formalization of "classical structures" in the `zxCalculus` (Rem. 4.15).

While — in formulating the quantum language `QS` below in §6 — we focus on language constructs for the Copenhagen-style typing (since this brings out the desired *dynamic lifting* of quantum-to-classical control, Lit. 2.9), the situation (150) shows that and how the ambient `LHoTT` language may in principle also be used to verify protocols in Everett-style formalisms such as the `zxCalculus`.

5 Quantum probability

5.1 Quantum probability from KR-Linearity

The discussion above captures all core aspects of quantum physics except the final postulate, the *Born rule*, which connects quantum physics to probability theory and hence to observable reality. Here we explain how the hermitian inner product structure and hence the probabilistic content of quantum state spaces arises from understanding quantum physics in KR-linear homotopy theory, where KR denotes the $\mathbb{Z}/2$-equivariant ring spectrum representing Atiyah's Real K-theory.

Hermitian structure via dependent linear types. So far we have quantum gates but no language structure to enforce their unitarity. While an inner product on a real vector space $\mathcal{V}$ may be axiomatized as an isomorphism $\mathcal{V} \xrightarrow{\sim} \mathcal{V}^*$ with the dual vector space, for sesquilinear inner products on a complex vector space the analogous isomorphism is (or would be) complex anti-linear, hence is not available as a morphism of $\mathbb{C}$-modules. Given a hermitian inner product $\langle -|-\rangle$ on a complex vector space $\mathcal{H}$, encoded by its induced anti-linear isomorphism given by hermitian conjugation, we may naturally consider the direct sum $\mathcal{H} \oplus \mathcal{H}^*$ as a $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-module $\underline{\mathcal{H}} \in \mathrm{PSh}(\mathbf{B}\mathbb{Z}_2)$ in the topos of sets equipped with $\mathbb{Z}_2$-actions:

<table>
<tr><th>Hermitian complex vector spaces ↪</th><th>$(\mathbb{Z}_2 \circlearrowright \mathbb{C})$-Modules in topos of involutions</th></tr>
<tr><td>

Hermitian inner product

$$\begin{tikzcd} \langle -|-\rangle : \mathcal{H} \times \mathcal{H} \longrightarrow \mathbb{C} \arrow[d, squiggly] \\ \textbf{hermitian conjugation} \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{H} \arrow[r, "(-)^\dagger", "\text{complex anti-linear}"'] & \mathcal{H}^* \arrow[r, "(-)^\dagger", "\text{complex anti-linear}"'] & \mathcal{H} \\ |\psi\rangle := \psi \arrow[r, mapsto] & \langle\psi|-\rangle =: \langle\psi| \arrow[r, mapsto] & |\psi\rangle \end{tikzcd}$$

</td><td>

$$\begin{tikzcd} \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] \otimes (\mathcal{H} \oplus \mathcal{H}^*) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "(-)\cdot(-)"] & \mathcal{H} \oplus \mathcal{H}^* \arrow[loop above, "\mathbb{Z}_2"] \end{tikzcd}$$

equivariant module structure

$$\begin{tikzcd} & \bullet \arrow[d, "C"] \arrow[dd, bend right=60, "\mathrm{id}"'] & & (c,\, |\psi\rangle) \arrow[r, mapsto] \arrow[d, mapsto] & c \cdot |\psi\rangle \arrow[d, mapsto, "(-)^\dagger"] \\ \underline{\mathcal{H}} : & \bullet \arrow[d, "C"] & \mapsto & (\overline{c},\, \langle\psi|) \arrow[r, mapsto] \arrow[d, mapsto] & \overline{c} \cdot \langle\psi| \arrow[d, mapsto, "(-)^\dagger"] \\ & \bullet & \mapsto & (c,\, |\psi\rangle) \arrow[r, mapsto] & c \cdot |\psi\rangle \end{tikzcd}$$

</td></tr>
<tr><td>

complex linear map

$A \;:\; \mathcal{H} \longrightarrow \mathcal{K}$

</td><td>

$$\begin{tikzcd} \underline{\mathcal{H}} \arrow[r, "\underline{A}"] & \underline{\mathcal{K}} \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{H} \oplus \mathcal{H}^* \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "A \oplus (A^\dagger)^*", "\textbf{hermitian adjoint operators}"'] & \mathcal{K} \oplus \mathcal{K}^* \arrow[loop above, "\mathbb{Z}_2"] \end{tikzcd}$$

$$\begin{tikzcd} \bullet \arrow[d, "C"] & & |\psi\rangle \arrow[r, mapsto] \arrow[d, mapsto] & A \cdot |\psi\rangle \arrow[d, mapsto] \\ \bullet & \mapsto & \langle\psi| \arrow[r, mapsto] & \langle\psi| \cdot A^\dagger \end{tikzcd}$$

</td></tr>
<tr><td>

$A \;:\; \mathcal{H} \longrightarrow \mathcal{K}$ is unitary ⇔

</td><td>

$$|\psi\rangle\langle\phi| \;\mapsto\; A|\psi\rangle\langle\phi|A^\dagger$$

$$\begin{tikzcd} \underline{\mathcal{H}} \otimes \underline{\mathcal{H}} \arrow[rr, "\underline{A} \otimes \underline{A}"] \arrow[dr, "\mathrm{ev}"'] & \circlearrowright & \underline{\mathcal{K}} \otimes \underline{\mathcal{K}} \arrow[dl, "\mathrm{ev}"] \\ & \underline{\mathbb{C}} & \end{tikzcd}$$

</td></tr>
</table>

(151)

For example, the complex numbers regarded as a 1d Hilbert space this way, look as follows:

$$\begin{tikzcd} & & & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] \otimes \big(\mathbb{C} \oplus \mathbb{C}^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r] & \big(\mathbb{C} \oplus \mathbb{C}^*\big) \arrow[loop above, "\mathbb{Z}_2"] \\ & \bullet \arrow[d, "C"] & & \big(c,\, (z, w)\big) \arrow[r, mapsto] \arrow[d, mapsto, "(\mathrm{id}, \iota)"] & (c \cdot v, c \cdot w) \arrow[d, mapsto, "\iota"] \\ \underline{\mathbb{C}} \;:\; & \bullet & \longmapsto & \big(\overline{c},\, (\overline{w},\, \overline{z})\big) \arrow[r, mapsto] & (\overline{c \cdot w},\, \overline{c \cdot v}) \end{tikzcd}$$

(152)

This induces the dagger involution

$$\begin{tikzcd}
\mathrm{Hom}\big(\underline{\mathcal{H}},\, \underline{\mathcal{K}}\big) \arrow[rr, "(-)^\dagger"] && \mathrm{Hom}\big(\underline{\mathcal{K}},\, \underline{\mathcal{H}}\big) \\
\underline{A} \,=\, A \oplus (A^\dagger)^* \arrow[rr, mapsto] && A^\dagger \oplus A^* \,=\, \underline{A}^\dagger
\end{tikzcd}$$

Complex structure	**Real subobjects**
$\begin{tikzcd} \underline{\mathcal{H}} \arrow[r, "\underline{\mathrm{i}}"] & \underline{\mathcal{H}} \end{tikzcd}$	$\begin{tikzcd} \textbf{fixed locus} & \text{inside} & \textbf{equalizer} & \text{of} & & \\ \mathrm{eq}(\underline{\mathrm{i}} \otimes \underline{\mathrm{i}},\, \sigma,\, \mathrm{id})^{\mathbb{Z}_2} \arrow[r, hook] & & \mathrm{eq}(\underline{\mathrm{i}} \otimes \underline{\mathrm{i}},\, \sigma,\, \mathrm{id}) \arrow[r, hook] & & \underline{\mathcal{H}} \otimes \underline{\mathcal{H}} \arrow[r, shift left=2, "\underline{\mathrm{i}} \otimes \underline{\mathrm{i}}{,}\ \textbf{imaginary rotation}"] \arrow[r, "\mathrm{id}" description] \arrow[r, shift right=2, "\sigma{,}\ \textbf{factor swapping}"'] & \underline{\mathcal{H}} \otimes \underline{\mathcal{H}} \end{tikzcd}$
$\begin{tikzcd} \mathcal{H} \oplus \mathcal{H}^* \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{i} \cdot \beta"] & \mathcal{H} \oplus \mathcal{H}^* \arrow[loop above, "\mathbb{Z}_2"] \end{tikzcd}$	$\begin{tikzcd} \mathrm{Herm}(\mathcal{H}) \arrow[rr, hook] & & \mathcal{H} \otimes \mathcal{H}^* \arrow[loop above, "\mathbb{Z}_2"] \\ \textbf{hermitian operators} & \text{among} & \textbf{linear operators} \end{tikzcd}$
$\begin{tikzcd} \bullet \arrow[d, "C"] & {\vert\psi\rangle} \arrow[r, mapsto] \arrow[d, mapsto, "(-)^\dagger"] & \mathrm{i} \cdot {\vert\psi\rangle} \arrow[d, mapsto, "(-)^\dagger"] \\ \bullet & \langle\psi{\vert} \arrow[r, mapsto] & -\mathrm{i} \cdot \langle\psi{\vert} \end{tikzcd}$	$\begin{tikzcd} \bullet \arrow[d, "C"] & \rho \arrow[d, equal] & {\vert\psi\rangle}\langle\phi{\vert} \arrow[d, mapsto, "(-)^\dagger"] \\ \bullet & \rho^\dagger & {\vert\phi\rangle}\langle\psi{\vert} \end{tikzcd}$

$$\begin{tikzcd}
& & \textbf{co-evaluation} & & & \textbf{evaluation} & \\
& \mathbb{C}_\bullet \arrow[r, "\underline{\mathrm{ev}}^\dagger"] & \underline{\mathbb{C}}_\bullet & \otimes & \underline{\mathbb{C}}_\bullet \arrow[r, "\underline{\mathrm{ev}}"] & \mathbb{C}_\bullet & \textbf{dependent 1d Hilbert spaces} \\
& & & & & & \text{modeled as} \\
& \mathbb{C}_\bullet \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}^\dagger_\bullet"] & \big(\mathbb{C}_\bullet \oplus \mathbb{C}^*_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] & \otimes & \big(\mathbb{C}_\bullet \oplus \mathbb{C}^*_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}_\bullet"] & \mathbb{C}_\bullet \arrow[loop above, "\mathbb{Z}_2"] & \textbf{equivariant linear dependent types} \\
& & & & & & \text{with} \\
c : C \quad \vdash & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}^\dagger"] & \big(\mathbb{C} \oplus \mathbb{C}^*\big) \arrow[loop above, "\mathbb{Z}_2"] & \otimes & \big(\mathbb{C} \oplus \mathbb{C}^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}"] & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] & \textbf{equivariant linear component types}
\end{tikzcd}$$

$$\begin{tikzcd}
& & {\vert c\rangle} \otimes \langle c{\vert} \arrow[dd, mapsto] \arrow[r, mapsto] & \langle c{\vert}c\rangle = 1 \arrow[dd, mapsto] \\
1_c \arrow[urr, mapsto] \arrow[drr, mapsto] & + & & \\
& & \langle c{\vert} \otimes {\vert c\rangle} \arrow[r, mapsto] & \langle c{\vert}c\rangle = 1 \\
& & {\vert c\rangle} \otimes {\vert c\rangle} \arrow[d, mapsto] \arrow[r, mapsto] & 0 \arrow[d, mapsto] \\
& & \langle c{\vert} \otimes \langle c{\vert} \arrow[r, mapsto] & 0
\end{tikzcd}$$

The key example is the type $\underline{\mathrm{QBit}}_\bullet$ of qbits with its hermitian inner product structure, obtained as the plain qbit type $\mathrm{QBit}_\bullet$ (**??**) but now with hermitian fibers $\underline{\mathbb{C}}$ (152):

$$\underline{\mathrm{QBit}}_\bullet \;:=\; (p_{\mathrm{Bool}})^* \underline{\mathbb{C}} \;:\; \mathbb{Z}_2\mathrm{LinType}_{\mathrm{Bool}}$$

This yields the 2-dimensional Hilbert space

$$\begin{tikzcd}
& & \underline{\mathbb{C}}^2 \,:=\, \underline{\mathbb{C}} \oplus \underline{\mathbb{C}} \\
\square_{\mathrm{Bool}}\big(\underline{\mathrm{QBit}}_\bullet\big) \;: & \bullet \arrow[d, "C"] & q_0{\vert 0\rangle} + q_1{\vert 1\rangle} \arrow[d, mapsto] \\
& \bullet & \overline{q_0}\langle 0{\vert} + \overline{q_1}\langle 1{\vert}
\end{tikzcd}$$

This works because of the Wirthmüller relation $\big(p_! \mathcal{H}\big)^* \,\simeq\, p_*\big(\mathcal{H}^*\big)$. We may thus add underlines to all the above discussion of quantum gates without changing their nature, but now allowing generalization to mixed states.

Complex anti-linear involutions are exactly what we get from CPT-equivariant structure if we work with $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-

modules: A $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-module $\mathbb{Z}_2 \circlearrowright \mathcal{V}$ is a complex vector space equipped with an ani-linear involution:

$$\begin{tikzcd}
 & \overset{\mathbb{Z}_2 \circlearrowright}{\mathbb{C}} \otimes \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}} \arrow[r] & \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}} \\
\bullet \arrow[d, "C"] & (c,\, v) \arrow[r, mapsto] \arrow[d, mapsto, "(\mathrm{id},\iota)"] & c \cdot v \arrow[d, mapsto, "\iota"] \\
\bullet & \big(\overline{c},\, \iota(v)\big) \arrow[r, mapsto] & \overline{c} \cdot \iota(v) \;=\; \iota(c \cdot v)
\end{tikzcd}$$

There are two kinds of complex structures on the underlying $\mathbb{R}$-module of such a $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-module (hence compatible actual $\mathbb{C}$-module structures, for $\mathbb{C}$ with its trivial involution action) amounting to a $\mathbb{Z}_2$-grading

(i) One is given by

$$\begin{tikzcd}
\overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_+ \oplus \mathcal{V}_-} \arrow[r, "\mathrm{i}\cdot\beta"] & \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_+ \oplus \mathcal{V}_-} \\
v \arrow[r, mapsto] \arrow[d, mapsto] & \mathrm{i} \cdot v \arrow[d, mapsto] \\
\iota(v) \arrow[r, mapsto] & -\mathrm{i} \cdot \iota(v)
\end{tikzcd}$$

(ii) the other by

$$\mathbb{Z}_2 \circlearrowright \mathcal{V} \;\simeq\; (\mathbb{Z}_2 \circlearrowright \mathcal{V}_+) \oplus (\mathbb{Z}_2 \circlearrowright \mathcal{V}_-)$$

$$\begin{tikzcd}
\overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_+} \otimes \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_-} \arrow[r, "\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}"] & \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_+} \otimes \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{V}_-}
\end{tikzcd}$$

In the former case, $\mathbb{Z}_2$ acts like charge-conjugation C, while in the latter case it acts like time-reversal T.

Example 5.1 ($\mathbb{C}$ as a 1d Hilbert space)**.**

$$\begin{tikzcd}
\underline{\mathbb{C}} & \overset{\mathbb{Z}_2 \circlearrowright}{\mathbb{C}} \otimes \overset{\mathbb{Z}_2 \circlearrowright}{\big(\mathbb{C} \oplus \mathbb{C}^*\big)} \arrow[r] & \overset{\mathbb{Z}_2 \circlearrowright}{\big(\mathbb{C} \oplus \mathbb{C}^*\big)} \\
\bullet \arrow[d, "C"] & \big(c,\, (z, w)\big) \arrow[r, mapsto] \arrow[d, mapsto, "(\mathrm{id},\iota)"] & (c \cdot v, c \cdot w) \arrow[d, mapsto, "\iota"] \\
\bullet & \big(\overline{c},\, (\overline{w},\, \overline{z})\big) \arrow[r, mapsto] & (\overline{c \cdot w},\, \overline{c \cdot v})
\end{tikzcd}$$

Given this, we may ask for the "real" subspace in the tensor square

$$\mathcal{V}_+ \otimes \mathcal{V}_- \,\oplus\, \mathcal{V}_- \otimes \mathcal{V}_+ \hookrightarrow \mathcal{V} \otimes \mathcal{V}\ .$$

Example 5.2. Examples of such $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-modules with complex structure are induced by complex Hermitian inner product spaces $\mathcal{H}$ as

$$\begin{tikzcd}
 & & & \overset{\mathbb{Z}_2 \circlearrowright}{\mathbb{C}} \otimes \overset{\mathbb{Z}_2 \circlearrowright}{(\mathcal{H} \oplus \mathcal{H}^*)} \arrow[r] & \overset{\mathbb{Z}_2 \circlearrowright}{\mathcal{H} \oplus \mathcal{H}^*} \\
 & & \bullet \arrow[d, "C"] & (\, c,\, |\psi\rangle\,) \arrow[r, mapsto] \arrow[d, mapsto] & c \cdot |\psi\rangle \arrow[d, mapsto, "(-)^\dagger"] \\
\underline{\mathcal{H}} & : & {} \arrow[r, phantom, "\mapsto"] & {} & {} \\
 & & \bullet & (\, \overline{c},\, \langle\psi|\,) \arrow[r, mapsto] & \overline{c} \cdot \langle\psi|
\end{tikzcd}$$

(i) This is of course a real structure on $\mathcal{H} \oplus \mathcal{H}^*$. A further real structure, amounting to a commuting action of $\{\mathrm{e}, C, P\}$, makes $\mathcal{H}$ a real Hilbert space. Here e is the neutral element.

(ii) These $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-modules are exactly those self-dual objects which have a *real* evaluation map (factoring through the real subspace, in the above sense), hence exactly the self-dual $(C, \mathrm{i}\beta)$-equivariant objects (if we regard $\mathbb{Z}_2 \circlearrowright \mathbb{C}$ as equipped with the trivial $\mathrm{i}\beta$-action):

$$\begin{tikzcd}
\big(\mathcal{H}\oplus\mathcal{H}^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \big(\mathcal{H}\oplus\mathcal{H}^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}"] & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] \\
|\psi_1\rangle \arrow[d, mapsto] & \langle\psi_2| \arrow[r, mapsto] & \langle\psi_2|\psi_1\rangle \arrow[d, mapsto] \\
\langle\psi_1| & |\psi_2\rangle \arrow[r, mapsto] & \langle\psi_1|\psi_2\rangle \\
|\psi_1\rangle \arrow[d, mapsto] & |\psi_2\rangle \arrow[r, mapsto] & 0 \arrow[d, mapsto] \\
\langle\psi_1| & \langle\psi_2| \arrow[r, mapsto] & 0
\end{tikzcd}$$

e.g.

$$\begin{tikzcd}
 & \big(\mathbb{C}_\bullet\oplus\mathbb{C}^*_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \big(\mathbb{C}_\bullet\oplus\mathbb{C}^*_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}"] & \mathbb{C}_\bullet \arrow[loop above, "\mathbb{Z}_2"] \\
c : C \ \vdash & |c\rangle \arrow[d, mapsto] & \langle c| \arrow[r, mapsto] & \langle c|c\rangle = 1 \arrow[d, mapsto] \\
 & \langle c| & |c\rangle \arrow[r, mapsto] & \langle c|c\rangle = 1 \\
 & |c\rangle \arrow[d, mapsto] & |c\rangle \arrow[r, mapsto] & 0 \arrow[d, mapsto] \\
 & \langle c| & \langle c| \arrow[r, mapsto] & 0
\end{tikzcd}$$

and a *real* coevaluation, given by

$$\begin{tikzcd}
\mathbb{C}_\bullet \arrow[loop above, "\mathbb{Z}_2"] \arrow[r] & \underline{\mathbb{C}}_\bullet \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \underline{\mathbb{C}}_\bullet \arrow[loop above, "\mathbb{Z}_2"]
\end{tikzcd}$$

$$\begin{tikzcd}
 & \mathbb{C}_\bullet \arrow[loop above, "\mathbb{Z}_2"] \arrow[r] & \big(\mathbb{C}_\bullet\oplus\mathbb{C}^*_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \big(\mathbb{C}^*_\bullet\oplus\mathbb{C}_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \\
c : C \quad \vdash & 1_c \arrow[r, mapsto] & |c\rangle & \langle c| \\
 & & +\ \langle c| & |c\rangle
\end{tikzcd}$$

(iii) A morphism between such $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-modules is a linear map A together with the dual of its hermitian adjoint $A^\dagger$

$$\begin{tikzcd}
 & \mathcal{H}_1\oplus\mathcal{H}_1^* \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "A\oplus\overline{A}"] & \mathcal{H}_2\oplus\mathcal{H}_2^* \arrow[loop above, "\mathbb{Z}_2"] \\
\bullet \arrow[d, "C"] & |\psi\rangle \arrow[r, mapsto] \arrow[d, mapsto] & A|\psi\rangle \arrow[d, mapsto] \\
\bullet & \langle\psi| \arrow[r, mapsto] & \langle\psi|A^\dagger
\end{tikzcd}$$

(iv) The tensor product of such a map with itself acts via Hermitian conjugation on the mixed terms and evaluation sends them to their inner product

$$\begin{tikzcd}
 & \big(\mathcal{H}_1\oplus\mathcal{H}_1^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \big(\mathcal{H}_1\oplus\mathcal{H}_1^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "(A\oplus\overline{A})\otimes(A\oplus\overline{A})"] & \big(\mathcal{H}_2\oplus\mathcal{H}_2^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, phantom, "\otimes"] & \big(\mathcal{H}_2\oplus\mathcal{H}_2^*\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[r, "\mathrm{ev}"] & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"] \\
\bullet \arrow[d, "C"] & |\psi\rangle & \langle\phi| \arrow[r, mapsto] \arrow[d, mapsto] & A|\psi\rangle & \langle\phi|A^\dagger \arrow[r, mapsto] \arrow[d, mapsto] & \langle\phi|A^\dagger A|\psi\rangle \arrow[d, mapsto] \\
\bullet & \langle\psi| & |\phi\rangle \arrow[r, mapsto] & \langle\psi|A^\dagger & A|\phi\rangle \arrow[r, mapsto] & \langle\psi|A^\dagger A|\phi\rangle
\end{tikzcd}$$

So the unitary maps are those maps of such $\mathbb{Z}_2 \circlearrowright \mathbb{C}$-modules whose tensor square is sliced over the evaluation map ev.

(...)

Mixed states/Density matrices via dependent linear types.

Given a Hermitian space $\mathcal{H}$, its density matrices form the subspace

$$\begin{tikzcd}
\mathrm{DMat}(\mathcal{H}) \arrow[r, hook] & \big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \otimes \big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \arrow[loop, out=130, in=100, "(-)^\dagger"] \arrow[loop, out=80, in=50, "\mathbb{Z}_2"] \arrow[loop below, "\mathrm{i}\cdot\beta"] \arrow[r, "\mathrm{ev}"] & \mathbb{C} \arrow[loop above, "\mathbb{Z}_2"]
\end{tikzcd}$$

which is the intersection of

- the reality condition: joint C- and β-fixed locus
- the trace=1 condition: fiber of ev over $1 \in \mathbb{C}$
- the positivity condition: existence of a real square root

$$\begin{tikzcd}
& \mathrm{States}(\mathcal{H}) \arrow[rr] \arrow[d] & & \{1\} \arrow[d, hook] \\
\big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \otimes \big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \arrow[r, shift left=2, "(-)^\dagger \otimes (-)^\dagger"] \arrow[r, "\mathrm{id}" description] \arrow[r, shift right=2, "\mathrm{i}\cdot\beta \otimes \mathrm{i}\cdot\beta"'] & \big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \otimes \big(\mathcal{H} \oplus \mathcal{H}^\ast\big) \arrow[rr, "\mathrm{ev}"] & & \mathbb{C}
\end{tikzcd}$$

Pure state preparation via density matrices:

$$\begin{tikzcd}
& \mathbb{C}_\bullet \arrow[r] & \big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big) \otimes \big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big) \arrow[r, "(\eta_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet)) \otimes (\eta_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet))"] & \Big(\Diamond_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \otimes \Big(\Diamond_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \\
c : C \quad \vdash & 1 \arrow[r, mapsto] & \begin{array}{cc} \vert c\rangle & \langle c\vert \\ + \quad \langle c\vert & \vert c\rangle \end{array} \arrow[r, mapsto] & \begin{array}{c} \vert c\rangle\langle c\vert \\ + \quad \langle c\vert\,\vert c\rangle \end{array}
\end{tikzcd} \tag{153}$$

condition that all such pure states are (semi-)positive as density matrices is equivalently the condition that the Hermitian inner product on $\mathcal{H}$ is (semi-)positive

measurement via density matrices

$$\begin{tikzcd}
& \big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big) \otimes \big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[dr, "\big(\eta_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet)\big) \otimes \big(\eta_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet)\big)"] & \\
\Big(\Box_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \otimes \Big(\Box_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \arrow[loop above, "\mathbb{Z}_2"] \arrow[ur, "\big(\epsilon_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet)\big) \otimes \big(\epsilon_C(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet)\big)"] \arrow[rr] & & \Big(\Box_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \otimes \Big(\Box_C\big(\mathcal{H}_\bullet \oplus \mathcal{H}^\ast_\bullet\big)\Big) \arrow[loop above, "\mathbb{Z}_2"]
\end{tikzcd}$$

$$\begin{tikzcd}
c : C \quad \vdash & \vert\psi\rangle \quad \langle\phi\vert \arrow[r, mapsto] \arrow[d, mapsto] & P_c\vert\psi\rangle \quad \langle\phi\vert P_c] \arrow[d, mapsto] \\
& \vert\phi\rangle \quad \langle\psi\vert \arrow[r, mapsto] & P_c\vert\phi\rangle \quad \langle\psi\vert P_c
\end{tikzcd}$$

or rather
comparison morphism

$$\underline{\mathcal{H}}_\bullet \otimes \underline{\mathcal{H}}_\bullet \xrightarrow{\eta_{\underline{\mathcal{H}}_\bullet} \otimes \eta_{\underline{\mathcal{H}}_\bullet}} \big((p_C)^*(p_C)_!\underline{\mathcal{H}}_\bullet\big) \otimes \big((p_C)^*(p_C)_!\underline{\mathcal{H}}_\bullet\big) \simeq (p_C)^*\big((p_C)_!\underline{\mathcal{H}}_\bullet \otimes (p_C)_!\underline{\mathcal{H}}_\bullet\big)$$

$$(p_C)_*\big(\underline{\mathcal{H}}_\bullet \otimes \underline{\mathcal{H}}_\bullet\big) \;\simeq (p_C)_!\big(\underline{\mathcal{H}}_\bullet \otimes \underline{\mathcal{H}}_\bullet\big) \xrightarrow{\widetilde{\eta_{\underline{\mathcal{H}}_\bullet} \otimes \eta_{\underline{\mathcal{H}}_\bullet}}} \big((p_C)_!\underline{\mathcal{H}}_\bullet\big) \otimes \big((p_C)_!\underline{\mathcal{H}}_\bullet\big) \simeq \; \big((p_C)_*\underline{\mathcal{H}}_\bullet\big) \otimes \big((p_C)_*\underline{\mathcal{H}}_\bullet\big)$$

$$\Box_C\big(\underline{\mathcal{H}}_\bullet \otimes \underline{\mathcal{H}}_\bullet\big) \xrightarrow{(p_C)^*\left(\widetilde{\eta_{\underline{\mathcal{H}}_\bullet} \otimes \eta_{\underline{\mathcal{H}}_\bullet}}\right)} \big(\Box_C\underline{\mathcal{H}}_\bullet\big) \otimes \big(\Box_C\underline{\mathcal{H}}_\bullet\big)$$

hence re-superposition as density matrices is:

$$\begin{tikzcd}
 & \Box_C\Big(\big(\Box_C\underline{\mathcal{H}}_\bullet\big) \otimes \big(\Box_C\underline{\mathcal{H}}_\bullet\big)\Big) \arrow[rr, "\substack{\textbf{tensor of}\\ \textbf{necessity counit}\\ \Box_C\left(\epsilon_{\underline{\mathcal{H}}_\bullet} \otimes \epsilon_{\underline{\mathcal{H}}_\bullet}\right)}"] & & \Box_C\big(\underline{\mathcal{H}}_\bullet \otimes \underline{\mathcal{H}}_\bullet\big) \arrow[ddr, "\substack{\textbf{necessity lax monoidalness}\\ (p_C)^*\left(\widetilde{\eta_{\underline{\mathcal{H}}_\bullet} \otimes \eta_{\underline{\mathcal{H}}_\bullet}}\right)}"] & \\
 & \bigoplus\limits_{c:C} \rho \arrow[rr, mapsto, "\textbf{collapse}"'] & & \bigoplus\limits_{c:C} P_c \cdot \rho \cdot P_c^\dagger \arrow[ddr, mapsto, "\substack{\textbf{embed pure states}\\ \textbf{among mixed}}"'] & \\
\big(\Box_C\underline{\mathcal{H}}_\bullet\big) \otimes \big(\Box_C\underline{\mathcal{H}}_\bullet\big) \arrow[uur, "\substack{\textbf{necessity co-operation}\\ (p_C)^*\left(\eta\left((p_C)_*\underline{\mathcal{H}}_\bullet \otimes (p_C)_*\underline{\mathcal{H}}_\bullet\right)\right)}"] & & & & \big(\Box_C\underline{\mathcal{H}}_\bullet\big) \otimes \big(\Box_C\underline{\mathcal{H}}_\bullet\big) \\
\overset{\textbf{mixed state}}{\rho} \arrow[uur, mapsto, "\textbf{parallelization}"'] \arrow[rrrr, mapsto, "\textbf{measurement quantum channel}"] & & & & \sum\limits_{c:C} P_c \cdot \rho \cdot P_c^\dagger
\end{tikzcd}$$

(...)
Better like this:

$$\begin{tikzcd}
B \arrow[rr, "\Delta"] \arrow[dr] & & B \times B \arrow[dl] \\
 & * &
\end{tikzcd}$$

$$☆_B \;=\; (p_B)_!(p_B)^* \;=\; (p_{B\times B})_!\Delta_!\Delta^*(p_{B\times B})^* \longrightarrow (p_{B\times B})_!(p_{B\times B})^* \;=\; ☆_{B\times B}$$

$$\bigcirc_B \;=\; (p_B)_*(p_B)^* \;=\; (p_{B\times B})_*\Delta_*\Delta^*(p_{B\times B})^* \longleftarrow (p_{B\times B})_*(p_{B\times B})^* \;=\; \bigcirc_{B\times B}$$

$$\Delta_!\Delta^*\Box_{B\times B}\mathbb{1}_{B\times B} \longrightarrow \Box_{B\times B}\mathbb{1}_{B\times B} \longrightarrow \mathbb{1}_{B\times B}$$

$$\bigcirc_{B\times B}\mathbb{1} \longrightarrow \bigcirc_B\bigcirc_{B\times B}\mathbb{1} \longrightarrow \bigcirc_{B\times B}\bigcirc_{B\times B}\mathbb{1} \longrightarrow \bigcirc_{B\times B}\mathbb{1}$$

$$\sum_{b,b'}\rho_{bb'} \cdot |b\rangle\langle b'| \quad \longmapsto \quad \bigoplus_{b''}\sum_{b,b'}\rho_{bb'} \cdot |b\rangle_{b''}\langle b|_{b''} \quad \longmapsto \quad \bigoplus_{b'',b'''}\delta_{b''.b'''}\sum_{b,b'}\rho_{bb'} \cdot |b\rangle_{b''}\langle b|_{b''} \quad \longmapsto \quad \sum_{b}\rho_{bb} \cdot |b\rangle\langle b|$$

6 QS Pseudocode

We now cast the categorical algebra of §3 and §4 into a programming-style language `QS` [20] that shall serve as pleasant but accurate pseudo-code for the actual encoding in `LHoTT`. Then we spell out a range of example programs.

`for...do`-Notation. The main language feature we use is standard "do-notation" for Kleisli maps, but sugared a little further in order to bring out a nicely intuitive quantum programming language. First, we write Kleisli maps for a monad $\mathcal{E}$ as "`for...do`" blocks in this somewhat non-standard form:

"`for...do...`" programming syntax for declaring effect-bound programs

$$\begin{array}{rcl} \text{prog} & : & D \to \mathcal{E}D' \\ \texttt{bind}^{\mathcal{E}}_{\text{prog}} & : & \mathcal{E}D \to \mathcal{E}D' \\ \texttt{bind}^{\mathcal{E}}_{\text{prog}} & \equiv & \left[\begin{array}{ll} \texttt{for} & \boxed{\texttt{return}^{\mathcal{E}}_{D}(d)} \\ \texttt{do} & \text{prog}(d) \end{array}\right. \end{array}$$

to be sugared as per next table

$$\begin{array}{lcl} \Phi \;:\; \mathcal{E}D, \quad \text{prog} \;:\; D \to \mathcal{E}D' & & \\ \phi \;\texttt{>}\; \texttt{bind}^{\mathcal{E}}_{\text{prog}} & : & \mathcal{E}D' \\ \phi \;\texttt{>}\; \texttt{bind}^{\mathcal{E}}_{\text{prog}} & \equiv & \left[\begin{array}{lll} \texttt{for} & \boxed{\texttt{return}^{\mathcal{E}}_{D}(b)} & \texttt{in} \;\; \Phi \\ \texttt{do} & \text{prog}(b) & \end{array}\right. \end{array}$$

This syntax is to closely reflect the fact that
– *for* an input of the form $\texttt{return}^{\mathcal{E}}_{D}(d) : \mathcal{E}D$,
– which may appear as a "summand" *in* the input data
– the operation $\texttt{bind}^{\mathcal{E}}_{\text{prog}}$ *does* produce the output prog(d),
which prescription completely defines it.

Beware that common classical notation for exactly the same construction is a little different:

$$\begin{array}{rcl} \texttt{bind}^{\mathcal{E}}_{\text{prog}} & : & \mathcal{E}D \to \mathcal{E}D' \\ \texttt{bind}^{\mathcal{E}}_{\text{prog}} & \equiv & \mathrm{e} \mapsto \left[\begin{array}{l} \texttt{do} \\ \quad d \leftarrow \mathrm{e} \\ \quad \text{prog}(d) \end{array}\right. \end{array}$$

This classical notation is meant to suggest that pure data $d : D$ may be "read out" from effectful data $e : \mathcal{E}D$. While this is suggestive for the list monad and its common relatives in classical programming, it is misleading in linear type theory and notably so for the quantum monad Q: Here the effectful input $e = |\psi\rangle$ is a quantum state like a q-bit, in which case d : Bit is a classical bit, whence the classical notation "$d \leftarrow |\psi\rangle$" could only be suggestive of performing a quantum measurement – in contradiction to the actual nature of the resulting $\texttt{bind}^{\mathrm{Q}}_{\text{prog}}$-operation constituting a coherent non-measurement quantum gate.

Instead, what really happens in Kleisli formalism is that operations are defined on *generators* for effectful data types $\mathcal{E}(D)$, namely on data of the form $\texttt{return}^{\mathcal{E}}_{D}(d)$. For example, the space of qbits $|\psi\rangle$: QBit is generated (here: linearly spanned) by the basis qbits $|0\rangle$ and $|1\rangle$, where we may naturally identify the ket-notation $|-\rangle$ as the unit/return operation which regards a classical bit b as the corresponding basis quantum state $|b\rangle$.

Proceeding in this vein, it is natural to declare the following syntactic sugar for the unit/return- and counit/extract-operations of all four potentia-modalities from **??**, according to the table further below.

---

[20] We call this language "`QS`", both as shorthand for "Quantum Systems Language" as well as alluding to the remarkable fact that (the semantics of) its universe of quantum data types goes far beyond the usual (Hilbert-) vector spaces to include "higher homotopy" linear types ("spectra"): Over the ground field $\mathbb{F}_1$, the quantization modality Q takes the circle homotopy type S to the "sphere spectrum" traditionally denoted "QS".

Sugared syntax for the (co)pure (co)monadic (co)effects

Quantization	$\vert - \rangle$	$\overset{\circ}{\circ}$	$B \to \mathrm{Q}B$	
	$\vert\, b \,\rangle$	$\equiv$	$\texttt{return}^{\mathrm{Q}}_{B}(b)$	pure linearity
Quantum measurement	`always`	$\overset{\circ}{\circ}$	$\mathscr{H} \multimap \bigcirc_B \mathscr{H}$	
	$\texttt{always}\, \vert\psi\rangle$	$\equiv$	$\texttt{return}^{\bigcirc_B}_{\mathscr{H}}\big(\vert\psi\rangle\big)$	pure indefiniteness
	`measure`	$\overset{\circ}{\circ}$	$\bigcirc_B \mathrm{Q}B \underset{B}{\multimap} \bigcirc_B \mathbb{1}$	
	$\texttt{measure}\, \vert\psi\rangle_b$	$\equiv$	$\texttt{obtain}^{\square_B}_{\mathbb{1}_B}\big(\vert\psi\rangle_b\big)$	pure necessity
	`measure`	$\overset{\circ}{\circ}$	$\mathrm{Q}B \multimap \bigcirc_B \mathbb{1}$	
	$\texttt{measure}\, \vert\psi\rangle$	$\equiv$	$\texttt{measure}\; \texttt{always}\, \vert\psi\rangle$	returns collapsed state & puts outcome into context
Quantum state preparation	`superpose`	$\overset{\circ}{\circ}$	$☆_B \mathscr{H} \multimap \mathscr{H}$	
	$\texttt{superpose}\, \vert\psi\rangle_b$	$\equiv$	$\texttt{obtain}^{☆_B}_{\mathscr{H}}\big(\vert\psi\rangle_b\big)$	pure randomness
	`prepare`	$\overset{\circ}{\circ}$	$☆_B \mathbb{1} \multimap ☆_B \mathrm{Q}B$	
	$\texttt{prepare}\, q_b$	$\equiv$	$\texttt{return}^{\Diamond_B}_{\mathbb{1}_B}\big(q_b\big)$	pure possibility
	`prepare`	$:$	$☆_B \mathbb{1} \multimap \mathrm{Q}B$	
	$\texttt{prepare}\, q_b$	$\equiv$	$\texttt{superpose}\; \texttt{prepare}\; q_b$	prepares states in context & returns superposed state

For example, with these conventions a linear map on QBit is coded by:

$$\Phi \;\overset{\circ}{\circ}\; \mathrm{QBit} \multimap \mathrm{QBit}$$

$$\Phi \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\texttt{do}\;\; \Phi|b\rangle \end{array}\right.$$

When nesting `for...do`-code we carry the argument using "`in`". For instance, given $I : D \multimap \mathrm{QBit}$, then its composite with Φ as above is:

$$I \;\texttt{>}\; \Phi \;\overset{\circ}{\circ}\; \quad D \multimap \mathrm{QBit}$$

$$I \;\texttt{>}\; \Phi \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \;\; \texttt{in} \;\; I \\ \;\texttt{do}\;\; \Phi|b\rangle \end{array}\right.$$

Similarly, while the tensor product is not a monad, it is also defined by generators whose value under linear maps uniquely defines these, and therefore we use the same `for...do`-notation for maps out of tensor products:

$$\begin{array}{lcl} \Phi & : & D \otimes D' \multimap E \\ \Phi & \equiv & \left[\begin{array}{l} \texttt{for}\ \ d \otimes d' \\ \ \ \texttt{do}\ \ \Phi(d \otimes d') \end{array}\right. \end{array} \qquad \begin{array}{lcl} \Phi & : & \mathrm{Q}B \otimes \mathrm{Q}B' \multimap E \\ \Phi & \equiv & \left[\begin{array}{l} \texttt{for}\ \ |b\rangle \otimes |b'\rangle \\ \ \ \texttt{do}\ \ \Phi\big(|b\rangle \otimes |b'\rangle\big) \end{array}\right. \end{array}$$

Typically, here d and d' are themselves effectful data types, in which case Φ may be coded by further nested `for...do`-loops, e.g.

$$\begin{array}{lcl} \Phi & : & \mathrm{QBit} \otimes \mathrm{QBit} \multimap \mathrm{QBit} \\ \Phi & \equiv & \left[\begin{array}{ll} \texttt{for}\ \ |\psi\rangle \otimes |\psi'\rangle & \\ \ \ \texttt{do} & \left[\begin{array}{l} \texttt{for}\ \ |b\rangle \otimes |b'\rangle\ \ \texttt{in}\ \ |\psi\rangle \otimes |\psi'\rangle \\ \ \ \texttt{do}\ \ \Phi\big(|b\rangle \otimes |b'\rangle\big) \end{array}\right. \end{array}\right. \end{array} \quad \text{abbreviated to} \quad \begin{array}{lcl} \Phi & : & \mathrm{QBit} \otimes \mathrm{QBit} \multimap \mathrm{QBit} \\ \Phi & \equiv & \left[\begin{array}{l} \texttt{for}\ \ |b\rangle \otimes |b'\rangle \\ \ \ \texttt{do}\ \ \Phi\big(|b\rangle \otimes |b'\rangle\big) \end{array}\right. \end{array}$$

Since Q is idempotent (in the relative sense: it is induced from an idempotent monad on Type), we may apply this notation in the generality that the codomain is any linear type, not necessarily explicitly of the form $\mathrm{Q}(-)$ (but always isomorphic to such).

This way, the strong monoidalness of $\mathrm{Q}(-)$ is witnessed by the following programs:

$$\begin{array}{l} \mathrm{Q}(\mathrm{Bit} \times \mathrm{Bit}) \multimap \mathrm{QBit} \otimes \mathrm{QBit} \\ \left[\begin{array}{l} \texttt{for}\ \ |b, b'\rangle \\ \ \ \texttt{do}\ \ |b\rangle \otimes |b'\rangle \end{array}\right. \end{array} \qquad \begin{array}{l} \mathrm{QBit} \otimes \mathrm{QBit} \multimap \mathrm{Q}(\mathrm{Bit} \times \mathrm{Bit}) \\ \left[\begin{array}{l} \texttt{for}\ \ |b\rangle \otimes |b'\rangle \\ \ \ \texttt{do}\ \ |b, b'\rangle \end{array}\right. \end{array}$$

Similarly, we introduce sugared syntax for the measurement monad:

The indefiniteness modality.

Quantum Measurement effects.

B : ClaType	**Quantum B-indefinitess modality** $\bigcirc_B$
general definition	$\bigcirc_B\ :\ \mathrm{Type} \to \mathrm{Type}$ $\bigcirc_B\ \equiv\ (p_B)_*(p_B)^*$ $\begin{tikzcd}[column sep=small] & B \times \natural\mathscr{E} \arrow[dl] \arrow[dr, "p_B"] & \\ B & & \natural\mathscr{E} \end{tikzcd}$
purely **classical** case	$\bigcirc_B\ :\ \mathrm{ClaType} \to \mathrm{ClaType}$ $\bigcirc_B\ \simeq\ \mathrm{id}$
purely **quantum** case	$\bigcirc_B\ :\ \mathrm{QuType} \to \mathrm{QuType}$ $\bigcirc_B\ \simeq\ \big(B \to (-)\big)$ strong wrt $\otimes$ if B : FinClaType **sugared syntax** for $\bigcirc_B$-data: $\big(B \to \mathscr{H}\big)\ \ \simeq\ \ \bigcirc_B \mathscr{H}$ $\big(b \mapsto \vert\psi_b\rangle\big)\ \ \mapsto\ \ \texttt{if measured}\ \ b\ \ \texttt{then}\ \ \vert\psi_b\rangle$

$$\begin{array}{lcl} \texttt{discard measurement} & : & \bigcirc_B \text{◌} \mathscr{H} \to \text{◌} \mathscr{H} \\ \texttt{proceed with} & \equiv & \texttt{return}^{\text{◌}} \end{array}$$

Proposition 6.1. *$\bigcirc_B$ a Frobenius monad, equivalent to the writer (co)monad of the (co)algebra $\mathbb{1}^B$. As such it coincides with Coecke's "classical structures" comonad.*

6.1 Standard q-bit circuit ingredients

$$\mathrm{X} \;⦂\; \mathrm{QBit} \multimap \mathrm{QBit}$$

$$\mathrm{X} \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\;\texttt{do}\;\; |b+1\rangle \end{array}\right.$$

$$\mathrm{H} \;⦂\; \mathrm{QBit} \multimap \mathrm{QBit}$$

$$\mathrm{H} \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\;\texttt{do}\;\; \tfrac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^{b}|1\rangle\Big) \end{array}\right.$$

$$\mathrm{CNOT} \;⦂\; \mathrm{Q}(\mathrm{Bit} \times \mathrm{Bit}) \multimap \mathrm{Q}(\mathrm{Bit} \times \mathrm{Bit})$$

$$\mathrm{CNOT} \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b_1,\, b_2\rangle \\ \;\;\texttt{do}\;\; |b_1,\; b_1 + b_2\rangle \end{array}\right.$$

6.2 Quantum Teleportation Protocol

```
Alice ⦂ $\mathrm{QBit} \multimap \big(\mathrm{QBit} \multimap \bigcirc_{\mathrm{Bit}\times\mathrm{Bit}} \mathbb{1}\big)$
Alice ≡
  for $|\mathrm{bell}_1\rangle$
  do
    for $|b\rangle$
    do $|b, \mathrm{bell}_1\rangle$ > $\mathrm{CNOT}$ > $(\mathrm{H} \otimes \mathrm{id})$ > measure
```

```
Bob ⦂ $\mathrm{QBit} \multimap \big(\bigcirc_{\mathrm{Bit}\times\mathrm{Bit}} \mathbb{1} \underset{\mathrm{Bit}\times\mathrm{Bit}}{\multimap} \bigcirc_{\mathrm{Bit}\times\mathrm{Bit}} \mathrm{QBit}\big)$
Bob ≡
  for $|\mathrm{bell}_2\rangle$
  do if measured $(b_1, b_2)$ then $|\mathrm{bell}_2\rangle$ > $\mathrm{X}^{b_1}$ > $\mathrm{Z}^{b_2}$
```

```
teleport ⦂ $\mathrm{QBit} \multimap \bigcirc_{\mathrm{Bit}\times\mathrm{Bit}} \mathrm{QBit}$
teleport ≡
  for $|b\rangle$
  do
    for $|\mathrm{bell}_1, \mathrm{bell}_2\rangle$ in ( prepare$(1_{0,0})$ > $(\mathrm{H} \otimes \mathrm{id})$ > $\mathrm{CNOT}$ )
    do $|b\rangle$ > $\mathrm{Alice}\big(|\mathrm{bell}_1\rangle\big)$ > $\mathrm{Bob}\big(|\mathrm{bell}_2\rangle\big)$
```

$$\text{verify} \quad : \quad \text{teleport} \;=\; \texttt{always} \qquad \text{i.e.} \qquad \prod_{|\psi\rangle : \mathrm{QBit}} \Big(\text{teleport}\,|\psi\rangle \;=\; \texttt{always}\,|\psi\rangle\Big)$$

Remark 6.2. Notice that the last expression provides the formal verification of the correct implementation of the teleportation protocol — and how the monadically typed `QS` (pseudo-)code is pleasantly close to a natural language rendering of this statement

> verify: "The quantum teleportation protocol applied to any quantum state $|\psi\rangle$ in Alice's hand *always* produces the same state $|\psi\rangle$ in Bob's machine, i.e. *independently of what measurement outcome* Alice happened to find in the process. In short: *The result of teleporting* $|\psi\rangle$ *is always* $|\psi\rangle$."

For analogous discussion of the verification in `LHoTT` of `Quipper`-code (Lit. 2.5) for quantum verification see [Ri23].

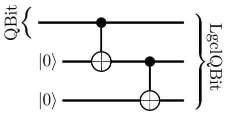
6.3 Quantum Bit Flip Code

Bit flip error correction as QS-pseudocode, is a simple but instructive example (cf. [NC10, §10.1.1]):

$\mathrm{LgclQBit} \;:\; \mathrm{QuType}$ $\qquad$ $\mathrm{Syndrome} \;:\; \mathrm{FinClaType}$

$\mathrm{LgclQBit} \;\equiv\; \mathrm{QBit}\otimes\mathrm{QBit}\otimes\mathrm{QBit}$ $\qquad$ $\mathrm{Syndrome} \;\equiv\; \mathrm{Bit}\times\mathrm{Bit}$

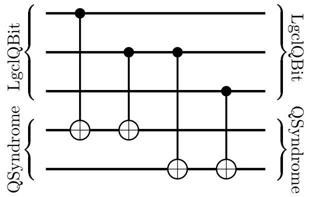
```
encode ⦂ $\mathrm{QBit} \multimap \mathrm{LgclQBit}$
encode ≡
    for $|b\rangle$
    do  $|b, b, b\rangle$
```



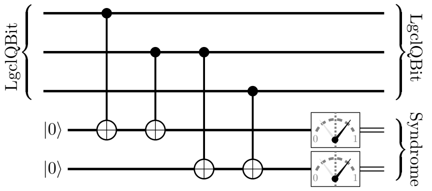
verify_circuit_encoding $\;:\;$ $\mathrm{encode} \;=\; (-)\otimes|0,0\rangle \;>\; \mathrm{CNOT}\otimes\mathrm{id} \;>\; \mathrm{id}\otimes\mathrm{CNOT}$

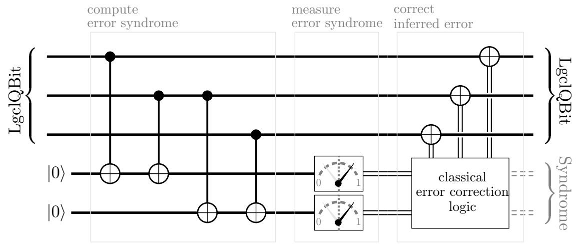
```
BitFlip ⦂ $\mathrm{Syndrome} \to (\mathrm{LgclQBit} \multimap \mathrm{LgclQBit})$
BitFlip ≡
    if $(0,0)$ then $\mathrm{id}\otimes\mathrm{id}\otimes\mathrm{id}$
    if $(1,0)$ then $\mathrm{X}\otimes\mathrm{id}\otimes\mathrm{id}$
    if $(1,1)$ then $\mathrm{id}\otimes\mathrm{X}\otimes\mathrm{id}$
    if $(0,1)$ then $\mathrm{id}\otimes\mathrm{id}\otimes\mathrm{X}$
```

```
compute_syndrome ⦂ $\mathrm{QSyndrome}\otimes\mathrm{LgclQBit} \multimap \mathrm{QSyndrome}\otimes\mathrm{LgclQBit}$
compute_syndrome ≡
    for $|s_1, s_2\rangle \otimes |b_1, b_2, b_3\rangle$
    do  $|s_1 + b_1 + b_2,\; s_2 + b_2 + b_3\rangle \otimes |b_1, b_2, b_2\rangle$
```



```
measure_syndrome ⦂ $\mathrm{LgclQBit} \multimap \bigcirc_{\mathrm{Syndrome}}\mathrm{LgclQBit}$
measure_syndrome ≡
    for $|b_1, b_2, b_3\rangle$
    do
        $|0,0\rangle \otimes |b_1, b_2, b_3\rangle$
        > compute_syndrome
        > measure$_{\mathrm{Syndrome}}$
```





```
correct_error ⦂ $\mathrm{LgclQBit} \multimap \bigcirc_{\mathrm{Syndrome}}\mathrm{LgclQBit}$
correct_error ≡
    for $|b_1, b_2, b_3\rangle$
    do
        for $|\psi\rangle$ in $\mathrm{measure\_syndrome}\big(|b_1, b_2, b_3\rangle\big)$
        do if measured $(s_1, s_2)$ then $\mathrm{BitFlip}_{(s_1,s_2)}|\psi\rangle$
```

verify_error_correction $\;:\;$ $(s_1, s_2 : \mathrm{Syndrome}) \to \Big(\mathrm{encode} \;>\; \mathrm{BitFlip}_{s_1,s_2} \;>\; \mathrm{correct\_error} \;=\; \texttt{always}\;\mathrm{encode}\Big)$

Remark 6.3. The last line asserts a term of identification type which *formally certifies* that any single bit flip on a logically encoded qbit is *always* corrected by the code (i.e.: no matter the measurement outcome). The construction of such certificates in `LHoTT` (not shown here, but straightforward in the present case) provides the desired formal verification of classically controlled quantum algorithms and protocols.

6.4 Repeat-Until-Success Quantum Gates

$$\text{Syndrome} \;:\; \text{FinClaType}$$

$$\text{Syndrome} \;\equiv\; \text{Bit}$$

$$\text{syndromed\_gate} \;⦂\; \text{QBit} \multimap \bigcirc_{\text{Syndrome}}\text{QBit}$$

```
syndromed_gate ≡
    for $|\psi\rangle$
    do $|0\rangle \otimes |\psi\rangle$ > Gate > $\texttt{measure}_{\text{Syndrome}}$
```

$$\text{uncompute} \;⦂\; \text{QBit} \multimap \text{QBit}$$

$$\text{uncompute} \;\equiv\; \text{whatever it takes}$$

$$\text{repeat\_until\_success} \;⦂\; \text{◌}\,\text{QBit} \multimap \text{◌}\,\text{QBit}$$

```
repeat_until_success ≡
    for $|\psi_{\text{in}}\rangle$
    do
        for $|\psi_{\text{out}}\rangle$ in $|\psi_{\text{in}}\rangle$ > syndromed_gate
        do
            if measured $0$ then proceed with $|\psi_{\text{out}}\rangle$
            if measured $1$ then
                proceed with uncompute$|\psi_{\text{out}}\rangle$
                > repeat_until_success
            discard measurement
```

$$\text{repeat\_until\_success} \;:\; \text{◌}\,\text{QBit} \xrightarrow{\text{◌}\,\text{syndromed\_gate}} \text{◌}\,\bigcirc_{B}\text{QBit} \xrightarrow{\text{◌}\,\text{control}} \text{◌}\,\bigcirc_{B}\text{◌}\,\text{QBit} \xrightarrow{\text{◌}\,\texttt{discard}} \text{◌}\,\text{◌}\,\text{QBit} \xrightarrow{\texttt{join}} \text{◌}\,\text{QBit}$$

7 Outlook

- we have not discussed formal `LHoTT` code here, but the translation is fairly straightforward, see [Ri23]

- the language `LHoTT` itself is not currently implemented in software the way `HoTT` is,
but there is no obstacle to such an implementation; and with the understanding of `LHoTT` as a universal quantum programming language there may now be the previously missing incentive for producing one.

- (...)

References

[Ab96] L. Abrams, *Two-dimensional topological quantum field theories and Frobenius algebra*, J. Knot. Theory and its Ramifications **5** (1996), 569-587, [`doi:10.1142/S0218216596000333`].

[Ab09] S. Abramsky, *No-Cloning in categorical quantum mechanics*, in: *Semantic Techniques for Quantum Computation*, Cambridge University Press (2009) 1-28, [`doi:10.1017/CBO9781139193313.002`], [`arXiv:0910.2401`].

[AD06] S. Abramsky and R. Duncan, *A Categorical Quantum Logic*, Mathematical Structures in Computer Science, **16** 3 (2006), 469-489, [`doi:10.1017/S0960129506005275`], [`arXiv:quant-ph/0512114`].

[Ac11] P. Aczel, *On Voevodsky's Univalence Axiom*, talk at the *Third European Set Theory Conference* (2011), [`ncatlab.org/nlab/files/Aczel-Univalence.pdf`]

[ACU15] T. Altenkirch, J. Chapman, and T. Uustalu, *Monads need not be endofunctors*, Logical Methods in Computer Science **11** 1:3 (2015) 1–40, [`doi:10.2168/LMCS-11(1:3)2015`], [`arXiv:1412.7148`].

[AC95] M. Adelman and J. Corbett, *A Sheaf Model for Intuitionistic Quantum Mechanics*, Appl. Cat. Struct. **3** (1995), 79-104, [`doi:10.1007/BF00872949`].

[ABGHR14] M. Ando, A. Blumberg, D. Gepner, M. Hopkins and C. Rezk, *An ∞-categorical approach to R-line bundles, R-module Thom spectra, and twisted R-homology*, J. Topology **7** 3 (2014) 869893, [`arXiv:1403.4325`], [`doi:10.1112/jtopol/jtt035`].

[AA22] R. W. Arroyo and J. R. B. Arenhart, *Whence deep realism for Everettian quantum mechanics?*, Foundations of Physics **52** 121 (2022), [`doi:10.1007/s10701-022-00643-0`], [`arXiv:2210.16713`].

[Ar$^+$15] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, *On the robustness of bucket brigade quantum RAM*, New J. Phys. **17** 12 (2015) 123010, [`arXiv:1502.03450`], [`doi:10.1088/1367-2630/17/12/123010`].

[Aw06] S. Awodey, *Category theory*, Oxford University Press (2010), [`doi:10.1093/acprof:oso/9780198568612.001.0001`].

[Aw12] S. Awodey, *Type theory and homotopy*, in: *Epistemology versus Ontology*, Springer, Berlin (2012), 183-201, [`doi:10.1007/978-94-007-4435-6_9`], [`arXiv:1010.1810`].

[AW09] S. Awodey and M. Warren, *Homotopy theoretic models of identity type*, Math. Proc. Cambridge Philosophical Society **146** (2009), 45-55, [`doi:10.1017/S0305004108001783`], [`arXiv:0709.0248`].

[Ba06] J. Baez, *Quantum Quandaries: a Category-Theoretic Perspective*, in D. Rickles et al. (eds.), *The structural foundations of quantum gravity*, Clarendon Press (2006), 240-265, [`ISBN9780199269693`], [`arXiv:quant-ph/0404040`]

[Ba96] A. Barber, *Dual Intuitionistic Linear Logic*, Technical Report ECS-LFCS-96-347, University of Edinburgh (1996) [`lfcs:96-347`]

[BBCDMSSSW95] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, *Elementary gates for quantum computation*, Phys. Rev. A **52** (1995) 3457-3467, [`doi:10.1103/PhysRevA.52.3457`], [`arXiv:quant-ph/9503016`].

[Be30] O. Becker, *Zur Logik der Modalitäten*, Jahrbuch für Philosophie und phänomenologische Forschung **11** (1930), 497-548, [`ophen:pub-101138`]; English translation: [`doi:10.1007/978-3-030-87548-0`].

[Bé67] J. Bénabou, *Introduction to Bicategories*, Lecture Notes in Mathematics **47**, Springer, Berlin (1967) 1-77 [`doi:10.1007/BFb0074299`].

[Bé85] J. Bénabou, *Fibered Categories and the Foundations of Naive Category Theory*, The Journal of Symbolic Logic **50** 1 (1985) 10-37 [`doi:10.2307/2273784`]

[BCR18] G. Benenti, G. Casati, and D. Rossini, *Principles of Quantum Computation and Information*, World Scientific, Singapore (2004, 2018), [`doi:10.1142/10909`].

[Be80] P. Benioff, *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines*, J. Stat. Phys. **22** (1980), 563–591, [`doi:10.1007/BF01011339`].

[Be95] N. Benton, *A mixed linear and non-linear logic: Proofs, terms and models*, in: *Computer Science Logic. CSL 1994*, Lecture Notes in Computer Science **933** (1995) 121-135 [`doi:10.1007/BFb0022251`]

[BBdP98] N. Benton, G. Bierman, and V. de Paiva, *Computational Types from a Logical Perspective*, J. Functional Programming **8** 2 (1998), 177-193, [`doi:10.1017/S0956796898002998`].

[BBdP92] N. Benton, G. Bierman and V. de Paiva, *Term assignment for intuitionistic linear logic*, Technical report **262**, Computer Laboratory University of Cambridge (August 1992) [`www.cl.cam.ac.uk/techreports/UCAM-CL-TR-262.pdf`]

[BBdPH92] N. Benton, G. Bierman, V. de Paiva and M. Hyland, *Linear λ-Calculus and Categorical Models Revisited*, in: *Computer Science Logic. CSL 1992*, Lecture Notes in Computer Science **702**, Springer (1993) [doi:10.1007/3-540-56992-8_6]

[BHM02] N. Benton, J. Hughes and E. Moggi, *Monads and Effects*, in: *Applied Semantics*, Lecture Notes in Computer Science **2395**, Springer (2002) 42-122 [`doi:10.1007/3-540-45699-6_2`]

[Bie07] G. Biedermann, private conversation with A. Joyal (2007).

[Bi94] G. Bierman, *On Intuitionistic Linear Logic*, Cambridge (1994) [`ncatlab.org/nlab/files/Bierman-LinearLogic.pdf`]

[BdP96] G. M. Bierman and V de Paiva, *Intuitionistic necessity revisited*, Birmingham CSR (1996), [`ncatlab.org/nlab/files/Bierman-dePaiva-NecessityRevisited.pdf`]

[BdP00] G. M. Bierman and V. de Paiva, *On an Intuitionistic Modal Logic*, Studia Logica **65** (2000), 383–416, [`doi:10.1023/A:1005291931660`].

[BvN36] G. Birkhoff and J. von Neumann, *The logic of quantum mechanics*, Ann. Math. **37** (1936), 823-843, [`doi:10.2307/1968621`].

[BvB07] P. Blackburn, J. van Benthem, *Modal logic: a semantic perspective*, Ch 1 in [BvBW07]: *The Handbook of Modal Logic*, Studies in Logic and Practical Reasoning **3** (2007), [`doi:10.1016/S1570-2464(07)80004-8`].

[BdRV01] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge Tracts in Theoretical Computer Science **53**, Cambridge University Press (2001), [`doi:10.1017/CBO9781107050884`].

[BvBW07] P. Blackburn, J. van Benthem, F. Wolter (eds.), *The Handbook of Modal Logic*, Studies in Logic and Practical Reasoning **3** (2007), [`https://cgi.csc.liv.ac.uk/~frank/MLHandbook`].

[Bo08] F. J. Boge, *Quantum Mechanics Between Ontology and Epistemology*, European Studies in Philosophy of Science (ESPS) **10**, Springer, Berlin (2008), [`doi:10.1007/978-3-319-95765-4`].

[Bohr1949] N. Bohr, *Discussion with Einstein on Epistemological Problems in Atomic Physics*, in: *Albert Einstein, Philosopher-Scientist*, The Library of Living Philosophers **VII** Evanston (1949) 201-241, *Niels Bohr Collected Works* **7** (1996), 339-381, [`doi:10.1016/S1876-0503(08)70379-7`].

[Bor94b] F. Borceux, *Handbook of Categorical Algebra* Vol. 2 *Categories and Structures*, Encyclopedia of Mathematics and its Applications **50** Cambridge University Press (1994), [`doi:10.1017/CBO9780511525865`].

[BEZ20] D. Bouwmeester, A. Ekert, and A. Zeilinger, *The Physics of Quantum Information – Quantum Cryptography, Quantum Teleportation, Quantum Computation*, Springer, New York (2020), [`doi:10.1007/978-3-662-04209-0`].

[BBDRV09] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, M. Van den Nest, *Measurement-based quantum computation*, Nature Physics **5** 1 (2009), 19-26, [`doi:10.1142/S0219749904000055`], [`arXiv:0910.1116`].

[Br14] J. W. Brunekreef, *Topological Quantum Computation and Quantum Compilation*, PhD thesis, Utrecht (2014), [`handle:20.500.12932/17738`].

[BLL13] G. Brunerie, D. Licata, and P. Lumsdaine, *Homotopy theory in type theory*, lecture notes (2013), [`dlicata.wescreates.wesleyan.edu/pubs/bll13homotopy/bll13homotopy.pdf`]

[Bu76] M. Bunge, *Possibility and Probability*, in: *Foundations of Probability Theory*, Statistical Interference, and Statistical Theories of Science, Reidel (1976), 17-34, [`doi:10.1007/978-94-010-1438-0_2`].

[BHI98] J. Butterfield, J. Hamilton, and C. Isham, *A topos perspective on the Kochen-Specker theorem, I. quantum states as generalized valuations*, Internat. J. Theoret. Phys. **37** 11 (1998), 2669-2733, [`doi:10.1023/A:1026680806775`].

[CJKP97] A. Carboni, G. Janelidze, M Kelly, and R. Paré, *On localization and stabilization for factorization systems*, Appl. Categ. Structures **5** (1997), 1-58, [`doi:10.1023/A:1008620404444`].

[CHK85] C. Cassidy, M. Hébert, M. Kelly, *Reflective subcategories, localizations, and factorization systems*, J. Austral. Math Soc. (Series A) **38** (1985) 287-329 [doi:10.1017/S1446788700023624]

[CR21] F. Cherubini and E. Rijke, *Modal Descent*, Math. Structures in Computer Science **31** 4 (2021), 363-391, [`doi:10.1017/S0960129520000201`], [`arXiv:2003.09713`].

[CW08] J. Clarke and F. K. Wilhelm, *Superconducting quantum bits*, Nature **453** (2008), 1031–1042, [`doi:10.1038/nature07128`].

[CS78] J. F. Clauser and A Shimony, *Bell's theorem: Experimental tests and implications*, Rep. Prog. Phys. **41** (1978) 1881, [`doi:10.1088/0034-4885/41/12/002`].

[Co23] B. Coecke, *Basic ZX-calculus for students and professionals*, [`arXiv:2303.03163`].

[CD08] B. Coecke and R. Duncan, *Interacting Quantum Observables*, in *Automata, Languages and Programming. ICALP 2008*, Lecture Notes in Computer Science **5126**, Springer (2008), [`doi:10.1007/978-3-540-70583-3_25`].

[CD11] B. Coecke and R. Duncan, *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*, New J. Phys. **13** (2011) 043016, [`doi:10.1088/1367-2630/13/4/043016`], [`arXiv:0906.4725`].

[CPac08] B. Coecke and E. O. Paquette, *POVMs and Naimark's theorem without sums*, Electronic Notes in Theoretical Computer Science **210** (2008) 15-31, [`doi:10.1016/j.entcs.2008.04.015`], [`arXiv:quant-ph/0608072`].

[CPP0909] B. Coecke, E. O. Paquette, and D. Pavlović, *Classical and quantum structuralism*, in: *Semantic Techniques in Quantum Computation*, Cambridge University Press (2009), 29-69, [`arXiv:0904.1997`], [`doi:10.1017/CBO9781139193313.003`].

[CPav08] B. Coecke and D. Pavlović, *Quantum measurements without sums*, in: *Mathematics of Quantum Computation and Quantum Technology*, Taylor & Francis (2008), 559-596, [`doi:10.1201/9781584889007`], [`arXiv:quant-ph/0608035`].

[CPV12] B. Coecke, D. Pavlović, and J. Vicary, *A new description of orthogonal bases*, Mathematical Structures in Computer Science **23** 3 (2012), 555-567, [`doi:10.1017/S0960129512000047`], [`arXiv:0810.0812`].

[CDL22] A. Colledan, U. Dal Lago, *On Dynamic Lifting and Effect Typing in Circuit Description Languages*, talk at *TYPES Workshop*, Nantes (2022), [`arXiv:2202.07636`].

[O'C11] R. O'Connor, *Functor is to Lens as Applicative is to Biplate: Introducing Multiplate*, contribution to ICFP '11: ACM SIGPLAN International Conference on Functional Programming (2011), [`arXiv:1103.2841`].

[Co11] R. Constable, *The Triumph of Types: Creating a Logic of Computational Reality*, lecture at: *Types, Semantics and Verification*, Oregon (2011), [`www.cs.uoregon.edu/research/summerschool/summer11/lectures/Triumph-of-Types-Extended.pdf`]

[Co20] D. Corfield, *Modal homotopy type theory*, Oxford University Press (2020), [`ISBN:9780198853404`].

[CC09] P. Cousot and R. Cousot, *A gentle introduction to formal verification of computer systems by abstract interpretation*, in: *Logics and Languages for Reliability and Security*, IOS (2009), [`doi:10.3233/978-1-60750-100-8-1`].

[DLF12] U. Dal Lago and C. Faggian, *On Multiplicative Linear Logic, Modality and Quantum Circuits*, EPTCS **95** (2012), 55-66, [`doi:10.4204/EPTCS.95.6`], [`arXiv:1210.0613`].

[DS22] S. Das Sarma, *Quantum computing has a hype problem*, MIT Technology Review (March 2022), [`technologyreview.com/2022/03/28/1048355/quantum-computing-has-a-hype-problem`].

[DN06] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, Quantum Information & Computation **6** 1 (2006), 81–95, [`doi:10.5555/2011679.2011685`], [`arXiv:quant-ph/0505030`].

[De89] D. E. Deutsch, *Quantum computational networks*, Proc. of the Royal Society A **425** 1868 (1989), 73-90, [`doi:10.1098/rspa.1989.0099`].

[dP89] V. de Paiva, *The Dialectica Categories*, in: *Categories in Computer Science and Logic*, Contemporary Mathematics **92** (1989) [`doi:10.1090/conm/092`]

[De14] S. J. Devitt, *Classical Control of Large-Scale Quantum Computers*, in: *Reversible Computation*, Lecture Notes in Computer Science **8507**, Springer, New York (2014), [`doi:10.1007/978-3-319-08494-7_3`], [`arXiv:1405.4943`].

[dWG73] B. S. DeWitt and N. Graham (eds.), *The Many-Worlds Interpretation of Quantum Mechanics*, Princeton Series in Physics, Princeton University Press (1973, 2016) [`ISBN:9780691645926`],[`jstor:j.ctt13x0wwk`].

[DHK08] H. Ditmarsch, W. Hoek, and B. Kooi, *Dynamic Epistemic Logic*, Studies In Epistemology, Logic, Methodology, And Philosophy Of Science (SYLI) **337** Springer, New York (2008), [`doi:10.1007/978-1-4020-5839-4`].

[Du06] R. Duncan, *Types for quantum mechanics*, PhD Dissertation. Oxford University (2006), [`personal.strath.ac.uk/ross.duncan/papers/rduncan-thesis.pdf`]

[EK66] S. Eilenberg and G. M. Kelly, *Closed Categories*, in: *Proceedings of the Conference on Categorical Algebra – La Jolla 1965*, Springer (1966) 421-562 [`doi:10.1007/978-3-642-99902-4`]

[EPR35] A. Einstein, B. Podolsky and N. Rosen, *Can the Quantum-Mechanical Description of Physical Reality be Considered Complete?* Physical Review **47** 10 (1935) 777-780 [doi:10.1103/PhysRev.47.777]

[Ev57a] H. Everett, *The Theory of the Universal Wavefunction*, PhD thesis, Princeton (1957), reprinted in [dWG73].

[Ev57b] H. Everett, *"Relative State" Formulation of Quantum Mechanics*, Rev. Mod. Phys. **29** (1957), 454-462, [`doi:10.1103/RevModPhys.29.454`].

[HM92] J. Y. Halpern and Y. Moses, *A guide to completeness and complexity for modal logics of knowledge and belief*, Artificial Intelligence **54** 3 (1992), 319-379, [`doi:10.1016/0004-3702(92)90049-4`].

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning About Knowledge*, The MIT Press (1995), [`ISBN:9780262562003`].

[FHTZ15] Y. Feng, E. M. Hahn, A. Turrini, and L. Zhang, `QPMC`*: A Model Checker for Quantum Programs and Protocols*, in *Formal Methods. FM 2015*, Lecture Notes in Computer Science **9109**, Springer, New York (2015), [`doi:10.1007/978-3-319-19249-9_17`].

[FG16] L. Ferroni and L. Gili, *Non-existent but potentially actual. Aristotle on plenitude*, Revue de philologie, de littérature et d'histoire anciennes, Tome XC (2016), 81-114, [`doi:10.3917/phil.901.0081`].

[Fey82] R. Feynman, *Simulating physics with computers*, Int J Theor Phys **21** (1982), 467–488, [`doi:10.1007/BF02650179`].

[Fey86] R. Feynman, *Quantum mechanical computers*, Found. Phys. **16** (1986), 507–531, [`doi:10.1007/BF01886518`].

[FSS23] D. Fiorenza, H. Sati, and U. Schreiber: *The Character map in Nonabelian Cohomology: Twisted, Differential and Generalized*, World Scientific, Singapore (2023), [`doi:10.1142/13422`], [`arXiv:2009.11909`], [`ncatlab.org/schreiber/show/The+Character+Map`],

[Fi07] M. Fitting, *Modal proof theory*, Ch. 2 in [BvBW07]: *The Handbook of Modal Logic*, Studies in Logic and Practical Reasoning **3** (2007), 85-183, [`doi:10.1016/S1570-2464(07)80005-X`].

[Fr99] P. Freyd, *Abelian-topos (AT) categories* (1999), [`ncatlab.org/nlab/show/AT+category`].

[FHPTST99] P. Freyd, P. O'Hearn, A. J. Power, M. Takeyama, R. Street, and R. D. Tennent, *Bireflectivity*, Theoretical Computer Science **228** 1–2 (1999), 49-76, [`doi:10.1016/S0304-3975(98)00354-5`].

[FKRS22] P. Fu, K. Kishida, N. J. Ross, and P. Selinger, *A biset-enriched categorical model for Proto-Quipper with dynamic lifting*, in *Quantum Physics and Logic 2022*, [`arXiv:2204.13039`].

[FKRS22] P. Fu, K. Kishida, N. J. Ross, and P. Selinger, *Proto-Quipper with dynamic lifting*, in *Quantum Physics and Logic 2022*, [`arXiv:2204.13041`].

[FKS20] P. Fu, K. Kishida and P. Selinger, *Linear Dependent Type Theory for Quantum Programming Languages*, LICS '20: Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (2020), 440–453, [`arXiv:2004.13472`].

[GLM08a] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum random access memory*, Phys. Rev. Lett. **100** (2008) 160501,[`doi:10.1103/PhysRevLett.100.160501`], [`arXiv:0708.1879`].

[GLM08b] V. Giovannetti, S. Lloyd and L. Maccone, *Architectures for a quantum random access memory*, Phys. Rev. A **78** (2008) 052310 , [`doi:10.1103/PhysRevA.78.052310`], [`arXiv:0807.4994`].

[Gir87] J.-Y. Girard, *Linear logic*, Theoretical Computer Science **50** (1987), 1-101, [`doi:10.1016/0304-3975(87)90045-4`].

[Gir93] J.-Y. Girard, *On the unity of logic* Annals of Pure and Applied Logic **59** (1993) 201-217 [`doi:10.1016/0168-0072(93)90093-S`]

[Gi03] R. Girle, *Possible Worlds*, McGill-Queen's University Press (2003), [`jstor:j.cttq48cx`].

[Gl09] J. Gleason, *The C^*-algebraic formalism of quantum mechanics* (2009), [`ncatlab.org/nlab/files/GleasonAlgebraic.pdf`]

[Gl11] J. Gleason, *From Classical to Quantum: The F^*-Algebraic Approach*, contribution to `VIGRE REU 2011`, Chicago (2011) [`ncatlab.org/nlab/files/GleasonFAlgebraic.pdf`]

[Go03] R. Goldblatt, *Mathematical modal logic: A view of its evolution*, J. Applied Logic **1** 5–6 (2003), 309-392, [`doi:10.1016/S1570-8683(03)00008-9`].

[GLLN08] J. Goubault-Larrecq, S. Lasota and D. Nowak, *Logical Relations for Monadic Types*, Mathematical Structures in Computer Science, **18** 6 (2008) 1169-1217 [`arXiv:cs/0511006`] [`doi:10.1017/S0960129508007172`]

[GLRSV13] A. Green, P. L. Lumsdaine, N. Ross, P. Selinger, and B. Valiron, *Quipper: A Scalable Quantum Programming Language*, ACM SIGPLAN Notices **48** 6 (2013), 333-342, [`doi:10.1145/2499370.2462177`], [`arXiv:1304.3390`].

[GLRSV13] A. Green, P. L. Lumsdaine, N. Ross, P. Selinger and B. Valiron, *An Introduction to Quantum Programming in Quipper*, Lecture Notes in Computer Science **7948**, Springer, New York (2013), 110-124, [`doi:10.1007/978-3-642-38986-3_10`], [`arXiv:1304.5485`].

[GRZ99] D. Greenberger, W. L. Reiter, and A. Zeilinger, *Epistemological and Experimental Perspectives on Quantum Physics*, Vienna Circle Institute Yearbook (VCIY) **7**, Springer, New York (1999), [`doi:10.1007/978-94-017-1454-9`].

[GMSW21] F. Groote, R. Morel, J. Schmaltz, and A. Watkins, *Logic Gates, Circuits, Processors, Compilers and Computers*, Springer, New York (2021), [`doi:10.1007/978-3-030-68553-9`].

[Gu95] C. A. Gunter, *The Semantics of Types in Programming Languages*, in: *Handbook of Logic in Computer Science*, Vol 3: *Semantic structures*, Oxford University Press (1995), [`ISBN:9780198537625`].

[GB22a] Y. Gurevich and A. Blass, *Quantum circuits with classical channels and the principle of deferred measurements*, Theor. Comp. Sci. **920** (2022), 21–32, [`doi:10.1016/j.tcs.2022.02.002`], [`arXiv:2107.08324`].

[Ha16] R. Harper, *Practical Foundations for Programming Languages*, Cambridge University Press (2016), [`ISBN:9781107150300`].

[Hed18] Hedera blog, *Formal Methods: The Importance of Being Fault Tolerant in a World with Bad Actors* (2018), [`medium.com/hedera:7308a4997fdd`].

[HSM$^+$20] B. Heim, M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, M. Troyer, and K. Svore, *Quantum programming languages*, Nature Reviews Phys. **2** (2020), 709–722, [`doi:10.1038/s42254-020-00245-7`].

[HLS02] C. Heunen, K. Landsman, and B. Spitters, *A topos for algebraic quantum theory*, Comm. Math. Phys. **291** (2009), 63-110, [`doi:10.1007/s00220-009-0865-6`], [`arXiv:0709.4364`].

[HV19] C. Heunen and J. Vicary, *Categories for Quantum Theory*, Oxford University Press (2019), [`ISBN:9780198739616`].

[HRHWH21] K. Hietala, R. Rand, S.-H. Hung, X. Wu, and M. Hicks, *A verified optimizer for Quantum circuits*, Proceedings of the ACM on Programming Languages **5** Issue POPL 37 (2021), 1–29, [`doi:10.1145/3434318`].

[HRHLH21] K. Hietala, R. Rand, S.-H. Hung, L. Li and M. Hicks, *Proving Quantum Programs Correct*, in: *12th International Conference on Interactive Theorem Proving (ITP 2021)*, Leibniz International Proceedings in Informatics (LIPIcs) **193** (2021), [`arXiv:2010.01240`].

[Hi62] K. J. J. Hintikka, *Knowledge and belief: An introduction to the logic of the two notions*, Cornell University Press (1962), [`ark:/13960/t9k437s65`].

[HP07] W. Hoek and M. Pauly, *Epistemic Logic*, Sec 4 in: *Modal logic for games and information*, Ch. 20 in [BvBW07]: *The Handbook of Modal Logic*, Studies in Logic and Practical Reasoning **3** (2007), 85-183, [`doi:10.1016/S1570-2464(07)80023-1`].

[Ho95] M. Hofmann, *Extensional concepts in intensional type theory*, Ph.D. thesis, Edinburgh (1995), Distinguished Dissertations, Springer (1997), [`ECS-LFCS-95-327`],[`doi:10.1007/978-1-4471-0963-1`].

[Ho97] M. Hofmann, *Syntax and semantics of dependent types*, in: *Semantics and logics of computation*, Publ. Newton Inst. **14**, Cambridge Univ. Press (1997), 79-130, [`doi:10.1017/CBO9780511526619.004`].

[HS98] M. Hofmann and T. Streicher, *The groupoid interpretation of type theory*, in: *Twenty-five years of constructive type theory*, Oxf. Logic Guides. **36**, Clarendon Press (1998) 83-111, [`ISBN:9780198501275`].

[HZBS07] L. Hormozi, G. Zikos, N. E. Bonesteel, and S. H. Simon, *Topological Quantum Compiling*, Phys. Rev. **B 75** (2007) 165310, [`doi:10.1103/PhysRevB.75.165310`], [`arXiv:quant-ph/0610111`].

[Ho19] M. Hoyois, *Topoi of parametrized objects*, Theory & App. Categ. **34** 9 (2019), 243-248, [`tac:34-09`], [`arXiv:1611.02267`].

[HWFZ20] H.-L. Huang, D. Wu, D. Fan, and X. Zhu, *Superconducting Quantum Computing: A Review*, Science China Information Sciences **63** 8 (2020), 1-32, [`doi:10.1007/s11432-020-2881-9`], [`arXiv:2006.10433`].

[HN19] T. Huckle and T. Neckel, *Bits and Bugs: A Scientific and Historical Review on Software Failures in Computational Science*, SIAM (2019) [`doi:10.1137/1.9781611975567`].

[HT95] H. Hu and W. Tholen, *Limits in free coproduct completions*, Journal of Pure and Applied Algebra **105** (1995) 277-291 [`doi:10.1016/0022-4049(94)00153-7`]

[HP07] M. Hyland and J. Power, *The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads*, Electronic Notes in Theor. Comp. Sci. **172** (2007), 437-458, [`doi:10.1016/j.entcs.2007.02.019`].

[Iq$^+$23] M. Iqbal et al, *Creation of Non-Abelian Topological Order and Anyons on a Trapped-Ion Processor*, [`arXiv:2305.03766`].

[Ja93] B. Jacobs, *Comprehension categories and the semantics of type dependency*, Theoret. Comput. Sci. **107** 2 (1993), 169-207, [`doi:10.1016/0304-3975(93)90169-T`].

[Ja94] B. Jacobs, *Semantics of weakening and contraction*, Annals of Pure and Applied Logic **69** 1 (1994) 73-106 [`doi:10.1016/0168-0072(94)90020-5`]

[Ja98] B. Jacobs, *Categorical Logic and Type Theory*, Studies in Logic and the Foundations of Mathematics **141**, Elsevier (1998) [`ISBN:978-0-444-50170-7`]

[Ja17] G. Jaeger, *Quantum potentiality revisited*, Phil. Trans. R. Soc. A **375** (2017) 20160390, [`doi:10.1098/rsta.2016.0390`].

[JT94] G. Janelidze and W. Tholen, *Facets of descent I*, Applied Categorical Structures **2** 3 (1994), 245-281, [`doi:10.1007/BF00878100`].

[JRW10] M. Johnson, R. Rosebrugh and R. Wood, *Algebras and Update Strategies*, Journal of Universal Computer Science **16** (2010) [`doi:10.3217/jucs-016-05-0729`]

[Jo02] P. Johnstone *Sketches of an Elephant – A Topos Theory Compendium* **1**, Oxford University Press (2002), [`ISBN:9780198534259`].

[Jo08] A. Joyal, *Notes on Logoi* (2008), [`ncatlab.org/nlab/files/JoyalOnLogoi2008.pdf`]

[KBS14] V. Kliuchnikov, A. Bocharov, and K. M. Svore, *Asymptotically Optimal Topological Quantum Compiling*, Phys. Rev. Lett. **112** (2014) 140504, `doi:10.1103/PhysRevLett.112.140504`], [`arXiv:1310.4150`]. [

[Kn96] E. Knill, *Conventions for quantum pseudocode*, Los Alamos Technical Report LA-UR-96-2724 (1996), [`doi:10.2172/366453`].

[Ko97] S. Kobayashi, *Monad as modality*, Theoretical Computer Science **175** 1 (1997), 29-74, [`doi:10.1016/S0304-3975(96)00169-7`].

[Ki08] A. Kissinger, *Graph Rewrite Systems for Classical Structures in †-Symmetric Monoidal Categories*, MSc thesis, Oxford (2008), [`ncatlab.org/nlab/files/Kissinger-CLassicalStructures.pdf`]

[Ki09] A. Kissinger, *Exploring a Quantum Theory with Graph Rewriting and Computer Algebra*, in: *Intelligent Computer Mathematics. CICM 2009*, Lecture Notes in Computer Science **5625** (2009), 90-105, [`doi:10.1007/978-3-642-02614-0_12`].

[Ko72] A. Kock, *Strong functors and monoidal monads*, Arch. Math. **23** (1972) 113–120 [`doi:10.1007/BF01304852`]

[Ko04] J. Kock, *Frobenius Algebras and 2d Topological Quantum Field Theories*, Cambridge U. Press (2004), [`doi:10.1017/CBO9780511615443`].

[Kr63] S. A. Kripke, *Semantical Analysis of Modal Logic I. Normal Modal Propositional Calculi*, Mathematical Logic Quaterly **9** 5-6 (1963), 67-96, [`doi:10.1002/malq.19630090502`].

[Ku05] G. Kuperberg, *A concise introduction to quantum probability, quantum mechanics, and quantum computation* (2005), [`ncatlab.org/nlab/files/Kuperberg-ConciseQuantum.pdf`]

[LS86] J. Lambek and P. J. Scott *Introduction to higher order categorical logic*, Cambridge Studies in Advanced Mathematics **7** (1986), [`ISBN:0-521-24665-2`].

[La17] K. Landsman, *Foundations of quantum theory – From classical concepts to Operator algebras*, Springer Open, New York (2017), [`doi:10.1007/978-3-319-51777-3`].

[Lau06] A. Lauda, *Frobenius algebras and ambidextrous adjunctions*, Theory and Applications of Categories **16** 4 (2006), 84-122, [`tac:16-04`], [`arXiv:math/0502550`].

[Law69a] W. Lawvere, *Adjointness in Foundations*, Dialectica **23** (1969) 281-296; reprinted in: Reprints in Theory and Applications of Categories **16** (2006), 1-16, [`tac:tr16`].

[Law69b] W. Lawvere, *Ordinal sums and equational doctrines*, in: *Seminar on Triples and Categorical Homology Theory*, Lecture Notes in Mathematics **80**, Springer, Berlin (1969), 141-155, [`doi:10.1007/BFb0083085`].

[Lee22] D. Lee, *Formal Methods for Quantum Programming Languages*, Paris Saclay (Dec 2022), [`hal:tel-03895847`].

[LPVX21] D. Lee, V. Perrelle, B. Valiron, and Z. Xu, *Concrete Categorical Model of a Quantum Circuit Description Language with Measurement*, Leibniz International Proceedings in Informatics **213** (2021) 51:1-51:20, [`doi:10.4230/LIPIcs.FSTTCS.2021.51`], [`arXiv:2110.02691`].

[LeS77] E. J. Lemmon with D. Scott, *An Introduction to Modal Logic – The "Lemmon Notes"*, Blackwell (1977), [`ark:/13960/t3gz25k3h`].

[LL32] C. I. Lewis, C. H. Langford, *Symbolic Logic* (1932), Dover (2000), [`archive:symboliclogic0000lewi_x3c4`].

[LB20] F. Leymann and J. Barzen, *The bitter truth about gate-based quantum algorithms in the NISQ era*, Quantum Sci. Technol. **5** (2020) 044007, [`doi:10.1088/2058-9565/abae7d`].

[Lü51] G. Lüders, *Über die Zustandsänderung durch den Meßprozeß*, Ann. Phys. **8** (1951), 322–328, [`doi:10.1002/andp.19504430510`]; *Concerning the state-change due to the measurement process*, Ann. Phys. **15** 9 (2006), 663-670, [`ncatlab.org/nlab/files/Lueders-StateChange.pdf`]

[Lu94] Z. Luo, *Computation and Reasoning – A Type Theory for Computer Science*, Clarendon Press, Oxford (1994), [`ISBN:9780198538356`].

[Lu17] J. Lurie, *Higher Algebra* (2017), [`https://www.math.ias.edu/~lurie/papers/HA.pdf`]

[ML71] S. MacLane, *Categories for the Working Mathematician*, Graduate Texts in Mathematics **5** Springer, Berlin (1971), [`doi:10.1007/978-1-4757-4721-8`].

[MLM92] S. Mac Lane and I. Moerdijk, *Sheaves in Geometry and Logic – A First Introduction to Topos Theory*, Springer (1992), [`doi:10.1007/978-1-4612-0927-0`].

[Mal23] C. Malkiewich, *A convenient category of parametrized spectra*, [`arXiv:2305.15327`].

[Ma76] E. G. Manes, *Algebraic Theories*, Springer, Berlin, (1976), [`10.1007/978-1-4612-9860-1`].

[Ma80] Y. I. Manin, *Computable and Uncomputable*, Sov. Radio (1980), published in: *Mathematics as Metaphor: Selected essays of Yuri I. Manin*, Collected Works **20**, AMS (2007), 69-77, [`ISBN:978-0-8218-4331-4`].

[Ma00] Y. I. Manin, *Classical computing, quantum computing, and Shor's factoring algorithm*, Séminaire Bourbaki exp. 862, Astérisque **266** (2000), 375-404, [`numdam:SB_1998-1999__41__375_0`], [`arXiv:quant-ph/9903008`].

[MMRP21] M. Maronese, L. Moro, L. Rocutto, and E. Prati, *Quantum Compiling*, in: *Quantum Computing Environments*, Springer, New York (2021), 39-74, [`doi:10.1007/978-3-030-89746-8_2`], [`arXiv:2112.00187`].

[ML71] P. Martin-Löf, *A Theory of Types*, unpublished note (1971), [`ncatlab.org/nlab/files/MartinLoef1971-ATheoryOfTypes.pdf`]

[ML75] P. Martin-Löf, *An intuitionistic theory of types: predicative part*, in: *Logic Colloquium '73*, Studies in Logic and the Foundations of Mathematics **80** (1975), 73-118, [`doi:10.1016/S0049-237X(08)71945-1`].

[ML82] P. Martin-Löf, *Constructive Mathematics and Computer Programming*, Studies in Logic and the Foundations of Mathematics **104** (1982), 153-175, [`doi:10.1016/S0049-237X(09)70189-2`].

[ML84] P. Martin-Löf (notes by G. Sambin of a series of lectures given in Padua in 1980), *Intuitionistic type theory*, Bibliopolis, Naples (1984), [`ncatlab.org/nlab/files/MartinLofIntuitionisticTypeTheory.pdf`]

[MaSi06] P. May and J. Sigurdsson, *Parametrized Homotopy Theory*, Mathematical Surveys and Monographs **132** AMS (2006), [`ams:surv-132`], [`arXiv:math/0411656`].

[McDU22] D. McDermott and T. Uustalu, *What Makes a Strong Monad?*, EPTCS **360** (2022) 113-133 [`arXiv:2207.00851`] [`doi:10.4204/EPTCS.360.6`]

[Me11] C. Meadows, *Program Verification and Security*, in *Encyclopedia of Cryptography and Security*, Springer, New York (2011), [`10.1007/978-1-4419-5906-5_863`].

[Me55] G. H. Mealy, *A method for synthesizing sequential circuits,* in *The Bell System Technical Journal*, vol. 34, no. 5, pp. 1045-1079, Sept. 1955, [`10.1002/j.1538-7305.1955.tb03788.x`].

[Me73] J. Mehra, *The quantum principle: Its interpretation and epistemology*, Dialectica **27** 2 (1973), 75-157, [`jstor:42968519`].

[Me09] P.-A. Melliés, *Categorical semantics of linear logic*, in: *Interactive models of computation and program behaviour*, Panoramas et synthéses **27** (2009) 1-196 [`ncatlab.org/nlab/files/Mellies-CategoricalSemanticsLinear.pdf`] [`smf.emath.fr/publications/semantique-c`

[Me89] N. D. Mermin, *What's Wrong with this Pillow?*, Physics Today **42** 4 (1989), 9–11, [`doi:10.1063/1.2810963`].

[MN13] D. Mihályi and V. Novitzká, *What about Linear Logic in Computer Science?*, Acta Polytechnica Hungarica **10** 4 (2013) 147-160 [`acta.uni-obuda.hu/Mihalyi_Novitzka_42.pdf`]

[Mi19] B. Milewski, *Category Theory for Programmers*, Blurb (2019) [`ISBN:9780464243878`], [`github.com/hmemcpy/milewski-ctfp-pdf/releases/download/v1.3.0/category-theory-for-programmers.pdf`]

[MZD20] A. Miranskyy, L. Zhang, and J. Doliskani, *Is Your Quantum Program Bug-Free?*, in *ICSE-NIER '20: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results* (2020), 29–32, [`doi:10.1145/3377816.3381731`], [`arXiv:2001.10870`].

[Mog89] E. Moggi, *Computational lambda-calculus and monads*, in: *Proceedings of the Fourth Annual Symposium on Logic in Computer Science* (1989), 14-23, [`doi:10.1109/LICS.1989.39155`].

[Mog91] E. Moggi, *Notions of computation and monads*, Inform. and Comput. **93** (1991), 55-92, [`doi:10.1016/0890-5401(91)90052-4`].

[Mo19] V. Moretti, *Fundamental Mathematical Structures of Quantum Theory*, Springer, New York (2019), [`doi:10.1007/978-3-030-18346-2`].

[TQP] D. J. Myers, H. Sati and U. Schreiber, *Topological Quantum Gates in Homotopy Type Theory*, [`arXiv:2303.02382`].

[Mye22] D. J. Myers, *Orbifolds as microlinear types in synthetic differential cohesive homotopy type theory*, [`arXiv:2205.15887`].

[NPW07] R. Nagarajan, N. Papanikolaou, and D. Williams, *Simulating and Compiling Code for the Sequential Quantum Random Access Machine*, Electronic Notes in Theoretical Computer Science **170** (2007), 101-124, [`doi:10.1016/j.entcs.2006.12.014`].

[*n*Lab14] nLab, *Necessity and possibility*, revision 1 (Nov 2014), [`ncatlab.org/nlab/revision/necessity+and+possibility/1`].

[Nie03] M. Nielsen, *Quantum computation by measurement and quantum memory*, Physics Letters A **308** (2003), 96–100, [`doi:10.1016/S0375-9601(02)01803-0`].

[NC10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2010), [`ISBN:9780511976667`].

[No02] D. Nolan, *Topics in the Philosophy of Possible Worlds*, Routledge, Boca Raton (2002), [`ISBN:9780415516303`].

[NPS90] B. Nordström, K. Petersson, and J. M. Smith, *Programming in Martin-Löf's Type Theory*, Oxford University Press (1990), [`cse.chalmers.se/research/group/logic/book`].

[Nui12] J. Nuiten, *Bohrification of local nets of observables*, Proceedings of *QPL 2011*, EPTCS **95** (2012), 211-218, [`doi:10.4204/EPTCS.95.15`], [`arXiv:1006.1432`].

[O'H03] P. O'Hearn, *On bunched typing*, J. of Functional Programming **13** 4 (2003), 747-796, [`doi:10.1017/S0956796802004495`].

[OP99] P. O'Hearn, D. J. Pym, *The Logic of Bunched Implications*, The Bulletin of Symbolic Logic **5** 2 (1999), 215-244, [`doi:10.2307/421090`].

[Om94] R. Omnés, *The Interpretation of Quantum Mechanics* Princeton University Press (1994), [`ISBN:9780691036694`].

[PPR19] Park, Petruccione and Rhee, *Circuit-Based Quantum Random Access Memory for Classical Data*, Scientific Reports **9** 3949 (2019) [`doi:10.1038/s41598-019-40439-3`]

[PB00] A. Pati and S. L. Braunstein, *Impossibility of deleting an unknown quantum state*, Nature **404** (2000), 164-165, [`doi:10.1038/404130b0`], [`arXiv:quant-ph/9911090`].

[PCG23] K. Phalak, A. Chatterjee, and S. Ghosh, *Quantum Random Access Memory For Dummies*, [`arXiv:2305.01178`].

[PP02] G. Plotkin and J. Power, *Notions of Computation Determine Monads*, in: Foundations of Software Science and Computation Structures FoSSaCS 2002, Lecture Notes in Computer Science **2303** Springer, New York (2002), [`doi:10.1007/3-540-45931-6_24`].

[PS12] K. Ponto and M. Shulman, *Duality and traces in indexed monoidal categories*, Theory and Applications of Categories **26** 23 (2012) [`arXiv:1211.1555`] [`tac:26-23`]

[Pr18] J. Preskill, *Quantum Computing in the NISQ era and beyond*, Quantum **2** (2018) 79, [`arXiv:1801.00862`], [`doi:10.22331/q-2018-08-06-79`].

[Pr22] J. Preskill, *The Physics of Quantum Information*, talk at *The Physics of Quantum Information*, 28th Solvay Conference on Physics (2022), [`arXiv:2208.08064`].

[Pr83] H. Primas, *Chemistry, Quantum Mechanics and Reductionism*, Springer, Berlin (1983), [`doi:10.1007/978-3-642-69365-6`].

[Py02] D. Pym, *The Semantics and Proof Theory of the Logic of Bunched Implications*, Applied Logic Series **26**, Springer, New York (2002), [`doi:10.1007/978-94-017-0091-7`].

[Py08] D. J. Pym, *Errata and Remarks for: "The Semantics and Proof Theory of the Logic of Bunched Implications"* (i.e., for [Py02]), Bath (2008), [`cantab.net/users/david.pym/BI-monograph-errata.pdf`]

[Re22] J. M. Renes, *Quantum Information Theory*, De Gruyter (2022), [`doi:10.1515/9783110570250`].

[RS20] M. Rennela and S. Staton, *Classical Control, Quantum Circuits and Linear Logic in Enriched Category Theory*, Logical Methods in Computer Science **16** (2020) LMCS:6192, [`doi:10.23638/LMCS-16(1:30)2020`], [`arXiv:1711.05159`].

[Ri21] F. Rios, *On a Categorically Sound Quantum Programming Language for Circuit Description*, Dalhousie University (2021) [`hdl:10222/80771`]

[Se16] P. Selinger, *Introduction to Quipper*, talk at QPL2016 (2016), [`youtu.be/59frzb__Eqo`]

[PZ19] J. Paykin and S. Zdancewic, *A HoTT Quantum Equational Theory*, QPL2019 (2019), [`arXiv:1904.04371`].

[Pr92] V. Pratt, *Linear logic for generalized quantum mechanics*, in *Proceedings of Workshop on Physics and Computation (PhysComp'92)* (1992), 166-180, [`doi:10.1109/PHYCMP.1992.615518`].

[Qu23] Quantstamp blog, *Applying lightweight formal methods and SAT solvers to build better blockchain applications* (2023)

[Ra18] R. Rand, *Formally Verified Quantum Programming*, PhD Dissertation, UPenn (2018), [`ediss:3175`].

[RPZ18] R. Rand, J. Paykin, and S. Zdancewic, *QWIRE Practice: Formal Verification of Quantum Circuits in Coq*, EPTCS **266** (2018), 119-132, [`arXiv:1803.00699`].

[Ra12] Ratkovic12] K. S. Ratkovic, *Morita theory in enriched context* (2012) [`arXiv:1302.2774`] [`hal:tel-00785301`]

[RS20] M. Rennela and S. Staton, *Classical Control, Quantum Circuits and Linear Logic in Enriched Category Theory*, Log. Meth. Comp. Sci. **16** 1 (2020), [`doi:10.23638/LMCS-16(1:30)2020`], [`arXiv:1711.05159`].

[RP11] E. Rieffel and W. Polak, *Quantum Computing – A gentle introduction*, MIT Press, Boston (2011), [`ISBN:9780262526678`].

[Ri22] E. Riehl, *On the ∞-topos semantics of homotopy type theory*, lecture at *Logic and higher structures*, CIRM (Feb. 2022), [`emilyriehl.github.io/files/semantics.pdf`]

[Rij18] E. Rijke, *Dependent type theory*, *Introduction to Homotopy Type Theory*, lecture notes, CMU (2018), [`andrew.cmu.edu/user/erijke/hott`].

[Rij23] E. Rijke, *Introduction to Homotopy Type Theory*, Cambridge University Press (in print), [`arXiv:2212.11082`].

[RSS20] E. Rijke, M. Shulman, and B. Spitters, *Modalities in homotopy type theory*, Log. Meth. Comp. Sci. **16** (2020) 1, [`doi:10.23638/LMCS-16(1:2)2020`], [`arXiv:1706.07526`].

[Ri22] M. Riley, *A Bunched Homotopy Type Theory for Synthetic Stable Homotopy Theory*, PhD Thesis, Wesleyan University (2022), [`doi:10.14418/wes01.3.139`].

[Ri23] M. Riley, *A Linear Dependent Type Theory with Identity Types* (2023), [`mvr.hosting.nyu.edu/pubs/translation.pdf`]

[RFL21] M. Riley, E. Finster, and D. R. Licata *Synthetic Spectra via a Monadic and Comonadic Modality* (2021), [`arXiv:2102.04099`].

[RS18] F. Rios and P. Selinger, *A Categorical Model for a Quantum Circuit Description Language*, EPTCS 266 (2018), 164-178, [`doi:10.4204/EPTCS.266.11`], [`arXiv:1706.02630`].

[Ro15] N. J. Ross, *Algebraic and Logical Methods in Quantum Computation*, PhD thesis, Dalhousie University (2015), [`arXiv:1510.02198`].

[Sa10] D. Samet, *S5 knowledge without partitions*, Synthese **172** (2010), 145–155, [`doi:10.1007/s11229-009-9469-0`].

[SS23a] H. Sati and U. Schreiber, *Anyonic defect branes in TED-K-theory*, Rev. Math. Phys **35** 06 (2023) 2350009, [`doi:10.1142/S0129055X23500095`], [`arXiv:2203.11838`].

[SS23b] H. Sati and U. Schreiber: *Anyonic topological order in TED K-theory* Rev. Math. Phys. **35** 03 (2023) 2350001, [`doi:10.1142/S0129055X23500010`], [`arXiv:2206.13563`].

[SS20] H. Sati and U. Schreiber, *Proper Orbifold Cohomology*, [`arXiv:2008.01101`].

[EoS] H. Sati and U. Schreiber, *Entanglement of Sections* [`ncatlab.org/schreiber/show/Entanglement+of+Sections`]

[IHH] H. Sati and U. Schreiber, *Introduction to Hypothesis H* [`ncatlab.org/schreiber/show/Introduction+to+Hypothesis+H`]

[Sau17] J. Sau, *A Roadmap for a Scalable Topological Quantum Computer*, Physics **10** 68 (2017), [`physics.aps.org/articles/v10/68`].

[Sche73] E. Scheibe, *The logical analysis of quantum mechanics*, Pergamon Press (1973), [`archive:logicalanalysiso0000sche`].

[Sch13] U. Schreiber, *Differential cohomology in an ∞-topos*, [`arXiv:1310.7930`].

[Sch14a] U. Schreiber, *Differential generalized cohomology in Cohesive homotopy type theory* , talk at Institut Henri Poincaré (2014), [`ncatlab.org/schreiber/files/SchreiberParis2014.pdf`]

[Sch14b] U. Schreiber, *Quantization via Linear Homotopy Types*, talks, [`arXiv:1402.7041`].

[Sch22] U. Schreiber, *Quantum Data Types via Linear Homotopy Type Theory* talk at *Workshop on Quantum Software @ QTML 2022*, Naples (Nov 2022), [`ncatlab.org/schreiber/files/QuantumDataInLHoTT-221117.pdf`]

[ScSh14] U. Schreiber and M. Shulman, *Quantum Gauge Field Theory in Cohesive Homotopy Type Theory*, EPTCS **158** (2014), 109-126, [`doi:10.4204/EPTCS.158.8`], [`arXiv:1408.0054`].

[Sc70] D. S. Scott, *Outline of a mathematical theory of computation*, in: Proc. 4th Ann. Princeton Conf. on Information Sciences and Systems (1970), 169–176, [`ncatlab.org/nlab/files/Scott-TheoryOfComputation.pdf`]

[ScSt71] D. S. Scott and C. Strachey, *Toward a Mathematical Semantics for Computer Languages*, Oxford Univ. Computing Laboratory, Technical Monograph PRG-6 (1971), [`cs.ox.ac.uk/files/3228/PRG06.pdf`]

[Se89] R. A. G. Seely, *Linear logic, $*$-autonomous categories and cofree coalgebras*, in: *Categories in Computer Science and Logic*, Contemporary Mathematics **92** (1989) [`ams:conm-92`]

[Se04] P. Selinger, *Towards a quantum programming language*, Math. Struc. Comp. Sci. **14** (2004), 527–586, [`doi:10.1017/S0960129504004256`].

[SV05] P. Selinger and B. Valiron, *A lambda calculus for quantum computation with classical control*, Proc. of TLCA (2005), [`arXiv:cs/0404056`].

[SV09] P. Selinger and B. Valiron, *Quantum Lambda Calculus*, in: *Semantic Techniques in Quantum Computation*, Cambridge University Press (2009), 135-172, [`doi:10.1017/CBO9781139193313.005`].

[Sel88] F. Selleri (ed.), *Quantum Mechanics Versus Local Realism – The Einstein-Podolsky-Rosen Paradox*, Physics of Atoms and Molecules (PAMO), Springer, Berlin (1988) [`doi:10.1007/978-1-4684-8774-9`].

[Sh94] P. W. Shor, *Algorithms for quantum computation: discrete logarithms and factoring*, Proc. 35th Annual Symposium on Foundations of Computer Science (1994), 124-134, [`doi:10.1109/SFCS.1994.365700`].

[Sh12] M. Shulman, *Minicourse on Homotopy Type Theory*, University of Swansea (2012), [`http://home.sandiego.edu/~shulman/hottminicourse2012`].

[Si97] D. R. Simon, *On the power of quantum computation*, SIAM J. Computing **26** 5 (1997), [`doi:10.1137/S0097539796298637`].

[Sip12] M. Sipser, *Introduction to the Theory Of Computation*, Cengage Learning (2012), [`ISBN:978-1-133-18779-0`].

[Sk76] B. Skyrms, *Possible Worlds, Physics and Metaphysics*, Philosophical Studies **30** 5 (1976), 323-332, [`jstor:4319099`].

[SK95] K. Slonneger and B. Kurtz, *Denotational semantics, Formal Syntax and Semantics of Programming Languages*, Addison-Wesley (1995), [`homepage.divms.uiowa.edu/ slonnegr/plf/Book/`].

[Spi19] D. Spivak, *Generalized Lens Categories via functors* $\mathcal{C}^{\mathrm{op}} \to \mathrm{Cat}$, [`arXiv:1908.02202`].

[Sta15] S. Staton, *Algebraic Effects, Linearity, and Quantum Programming Languages*, Proceedings of the 42nd Symposium on Principles of Programming Languages (2015), 395–406, [`doi:10.1145/2676726.2676999`].

[Str04] R. Street, *Frobenius monads and pseudomonoids*, J. Math. Phys. **45** (2004), 3930–3948, [`doi:10.1063/1.1788852`].

[St93] T. Streicher, *Investigations into Intensional Type Theory*, Habilitation Thesis, Darmstadt (1993), [`ncatlab.org/nlab/files/Streicher-IntensionalTT.pdf`]

[Ta00] P. Tappenden, *Identity and Probability in Everett's Multiverse*, British J. Philosophy Science **51** 1 (2000), 99-114, [`jstor:3541750`].

[Te98] M. Tegmark, *The Interpretation of Quantum Mechanics: Many Worlds or Many Words?*, Fortsch. Phys. **46** (1998), 855-862, [`arXiv:quant-ph/9709032`].

[Ter19] V. Terekhovich, *Modal Approaches in Metaphysics and Quantum Mechanics*, [`arXiv:1909.10046`].

[Th91] S. Thompson, *Type Theory and Functional Programming*, Addison-Wesley (1991), [`ISBN:0201416670`].

[UFP13] Univalent Foundations Project, *Homotopy Type Theory – Univalent Foundations of Mathematics*, Institute for Advanced Study, Princeton, 2013, [`homotopytypetheory.org/book`].

[Uu21] T. Uustalu, *Monads and Interaction*, lecture at *Midlands Graduate School in the Foundations of Computing Science* (2021), [`ncatlab.org/nlab/files/Uustalu-Monads1.pdf`],[`2.pdf`],[`3.pdf`],[`4.pdf`]

[Val04] B. Valiron, *A functional programming language for quantum computation with classical control*, MSc thesis, University of Ottawa (2004), [`doi:10.20381/ruor-18372`].

[VRSAS15] B. Valiron, N. J. Ross, P. Selinger, D. S. Alexander, and J. M. Smith, *Programming the quantum future*, Communications of the ACM **58** 8 (2015), 52–61, [`doi:10.1145/2699415`].

[vWe] J. van de Wetering, *ZX-calculus for the working quantum computer scientist*, [`arXiv:2012.13966`].

[vW51] G. H. von Wright, *An Essay in Modal Logic*, North-Holland Publishing, Amsterdam (1951), [`philpapers:VONAEI-2`].

[VYC22] S. Verma, D. Yadav, and G. Chandra, *Introduction of Formal Methods in Blockchain Consensus Mechanism and Its Associated Protocols*, IEEE Access **10** (2022), [`doi:10.1109/ACCESS.2022.3184799`].

[Vo10] V. Voevodsky, *Univalent Foundations Project*, grant proposal application (2010), [`ncatlab.org/nlab/files/Voevodsky-UFP2010.pdf`]

[vonNeumann1932] J. von Neumann, *Mathematische Grundlagen der Quantenmechanik* (1932, 1971), [`doi:10.1007/978-3-642-96048-2`]; *Mathematical Foundations of Quantum Mechanics*, Princeton University Press (1955), [`doi:10.1515/9781400889921`].

[Wa93] P. Wadler, *A syntax for linear logic*, in: *Ninth International Coference on the Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science **802** Springer (1993) [`doi:10.1007/3-540-58027-1_24`]

[Wei21] T.-C. Wei, *Measurement-Based Quantum Computation*, Oxford Research Encyclopedia of Physics (2021), [`doi:10.1142/S0219749904000055`], [`arXiv:2109.10111`].

[Wi20] A. Wilson, *The Nature of Contingency: Quantum Physics as Modal Realism*, Oxford University Press (2020), [`ISBN:9780198846215`].

[WLBF09] J. Woodcock, P. G. Larsen, J. Bicarregui, and J. Fitzgerald, *Formal methods: Practice and experience*, ACM Computing Surveys **41** 19 (2009), 1–36, [`doi:10.1145/1592434.1592436`].

[WZ] W. Wooters and W. Zurek, *A single quantum cannot be cloned*, Nature **299** (1982), 802-803, [`doi:10.1038/299802a0`].

[Ya19] R. Yates, *Improving Haskell Transactional Memory* (2019), [`hdl:1802/35367`].

[Ye90] D. Yetter, *Quantales and (noncommutative) linear logic*, J. Symbolic Logic **55** (1990), 41-64, [`doi:10.2307/2274953`].

[YF18] M. Ying and Y. Feng, *Model Checking Quantum Systems – A Survey*, [`arXiv:1807.09466`].

[YF21] M. Ying and Y. Feng, *Model Checking Quantum Systems – Principles and Algorithms*, Cambridge University Press (2021), [`ISBN:9781108484305`].

[Yu12] Q. Yuan, *Finite noncommutative probability, the Born rule, and wave function collapse* (2012), [`qchu.wordpress.com/2012/09/09/`].

[ZBSLY23] L. Zhou, G. Barthe, P.-Y. Strub, J. Liu, and M. Ying, `CoqQ`*: Foundational Verification of Quantum Programs*, Proceedings of the ACM on Programming Languages **7** POPL 09 (2023) 29, 833–865, [`doi:10.1145/3554343`], [`arXiv:2207.11350`].