The Quantum Monadology

Hisham Sati*† Urs Schreiber*

March 27, 2025

Abstract

The modern theory of functional programming languages uses monads for encoding computational side-effects and side-contexts, beyond bare-bone program logic. Even though quantum computing is intrinsically side-effectful (as in quantum measurement) and context-dependent (as on mixed ancillary states), little of this monadic paradigm has previously been brought to bear on quantum programming languages.

Here we systematically analyze the (co)monads on categories of parameterized module spectra which are induced by Grothendieck's "motivic yoga of operations" – for the present purpose specialized to $H\mathbb{C}$-modules and further to set-indexed complex vector spaces, as discussed in a companion article [SS23-EoS]. Interpreting an indexed vector space as a collection of alternative possible quantum state spaces parameterized by quantum measurement results, as familiar from Proto-`Quipper`-semantics, we find that these (co)monads provide a comprehensive natural language for functional quantum programming with classical control and with "dynamic lifting" of quantum measurement results back into classical contexts.

We close by indicating a domain-specific quantum programming language (`QS`) expressing these monadic quantum effects in transparent `do`-notation, embeddable into the recently constructed *Linear Homotopy Type Theory* (`LHoTT`) which interprets into parameterized module spectra. Once embedded into `LHoTT`, this should make for formally verifiable universal quantum programming with linear quantum types, classical control, dynamic lifting, and notably also with topological effects (as discussed in the companion article [TQP]).

Extended Abstract

Concretely, for *finite* classical and *finite-dimensional* quantum types (as of concern in quantum information theory), linear base change and linear internal hom constitute two ambidextrous adjunctions inducing a system of Frobenius monads which are linear/quantum versions of the classical Environment-, State-, and Epistemic-monads. We find that:

(i) The QuantumEpistemic modality neatly encodes the logic of controlled quantum gates.
  - Its Kleisli equivalence formally proves the deferred measurement principle.

(ii) The QuantumEnvironment monad coincides with Coecke's "classical structures" monad used in `zxCalculus`.
  - Its effect-handling computationally encodes collapsing quantum measurement "dynamically lifted" into the classical context akin to D. Lee's "lifting monad".
  - Its monoidal structure encodes enhancement of parameterized quantum circuits to mixed states.

(iii) The QuantumState monad produces spaces of density matrices.
  - Its monad transformations encode quantum channels acting on mixed quantum states.

Moreover, the QuantumEnvironment and QuantumState (co)monads pairwise distribute over each other as to provide a pair of 2-sided Kleisli categories, where:

- QuantumEnvironment-contextful and QuantumState-effectful maps encode mixed state preparation,
- on which QuantumState-transformations act as quantum channels, followed by
- QuantumState-contextful and QuantumEnvironment-effectful maps, encoding measurement and observables.

Notably, the action of QuantumState-transformations on QuantumState-contextful scalars (observables) is precisely Heisenberg-picture quantum evolution.

Finally, the QuantumEnvironment lifts from a monad on linear types to a (relative) monad on, in turn, QuantumState-monads, whereby the quantum effect logic for parameterized quantum circuits in the generality of mixed states becomes verbatim that for pure states, while mixed state effects such as the Born rule are brought out by the rich monadic semantics.

---

* 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`.

Contents

0.1 Motivation

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. 1.1) 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. 1.2), 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. 1.3) given by topological quantum materials (Lit. 1.23) whose registry-states are protected by an "energy gap" from having *any* interaction with the environment below that range.
- **(ii) Verified software** (Lit. 1.4) 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.

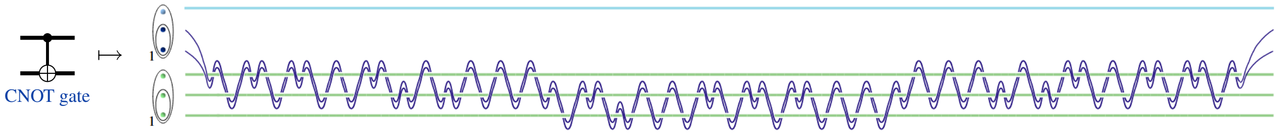
Both of these issues have been discussed separately, but the necessary combination has remained essentially untouched until [TQP]; one will need a quantum programing language (Lit. 1.5) which is

- **(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:

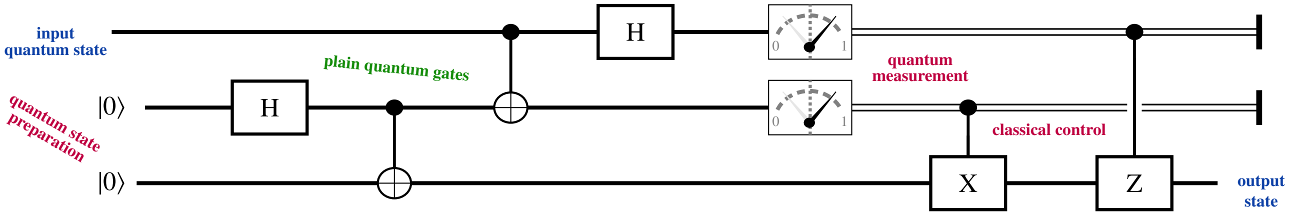
> **Topological quantum circuit compilation problem** (Lit. 1.9).
> *Suppose a topologically ordered quantum material is finally developed which features* su_2*-anyon states at level* ℓ*, and given any quantum circuit written in the usual QBit-basis, then the quantum compilation of this circuit onto 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 (17) may compile to the following braid [HZBS07, Fig. 15]:



Hence future quantum programmers will 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. §3.2.2):



---

[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. 1.10), serious quantum computation is expected (Lit. 1.11) to involve a perpetual loop of classical control operations on the quantum computer (*hybrid* classical/quantum computation). These are predominantly for quantum error correction (§3.2.3) but also for purposes such as repeat-until-success gates – where subsequent quantum circuit execution is classically conditioned on run-time quantum measurement results – also called "dynamic lifting" (Lit. 1.11, namely of quantum measurement results into the classical data register). This is schematically indicated on the right.
Last not least, for probabilistic analysis of such hybrid processes the machine state is to be modeled as a *mixed* classical/quantum probabilistic state (Lit. 1.12).

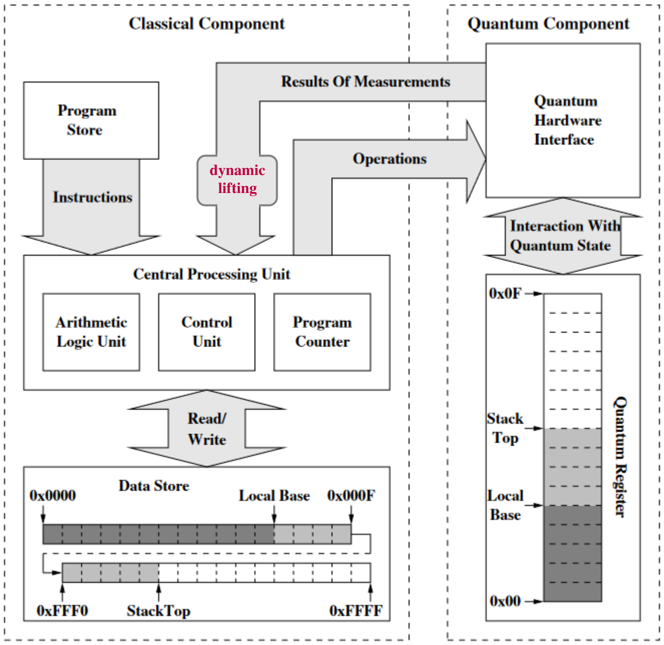
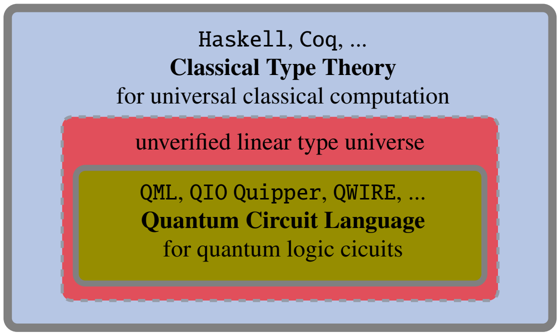


diagram adapted from [NPW07, Fig. 1]

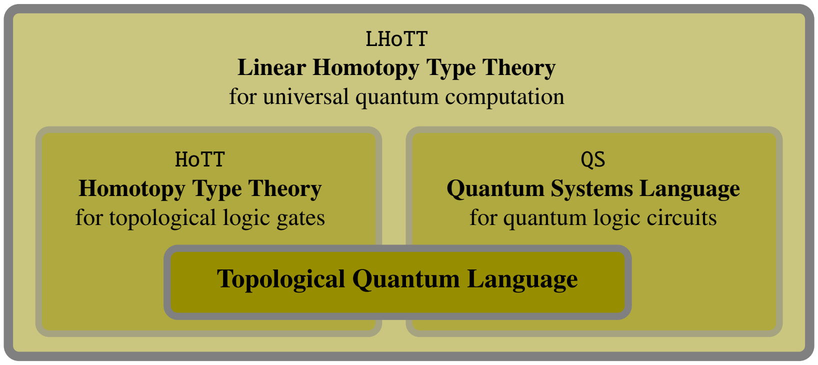
Hence for reliable heavy-duty quantum computation we need a certification language that knows about classical data types *and* about linear/quantum data types *and* their *dependency* on classical data. This had been lacking:

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 (Lit. 1.5) which do not have native support for linear types (cf. Lit. 1.4) nor for classical control of quantum circuits. For instance, basic protocol schemes such as quantum teleportation (§3.2.2), quantum error correction (§3.2.3) or repeat-until-success gates remain unverifiable with previous technology.

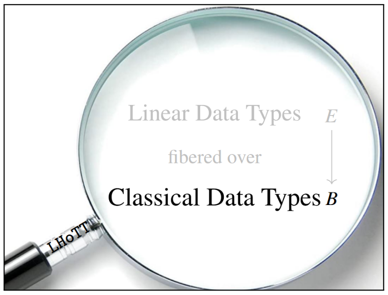


Solution by Linear Homotopy Type Theory. We argue here, as announced in [Sch22], that the novel type theory `LHoTT` (Lit. 1.8) recently developed in [Ri22a] (anticipated in [Sch14a]) in extension of the classical language scheme `HoTT` (Lit. 1.7) serves as the missing universal quantum programming/certification language.
Our claim is that `LHoTT`:

- Solves the old problem of constructing combined classical/linear type theories (cf. Lit. 1.4).
- 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 (§2).
- Natively supports verification of realistic topological quantum gates [TQP].



We argue that this makes `LHoTT` the first comprehensive paradigm for serious quantum programming beyond the NISQ area. Here we describe a domain-specific language embeddable into `LHoTT` to bring this out: *Quantum Systems Language* (`QS`, §3), based on a system of monadic effects which are definable (by admissible inference rules) in `LHoTT` (§2, surveyed below in §0.2).



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.

Quantum halos. Formally this is achieved by 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. 1.14) are the *purely classical* (ordinary) homotopy types, embedded *bi-reflectively* (157) among all data types (see §2.1):

> The presence of the $\natural$-modality exhibits general types E : Type as microscopic/infinitesimal *halos* around their underlying purely classical type $\natural E$: ClaType. It is a profound fact (146) of ∞-topos theory that models for such *infinitesimal cohesion* (see Lit. 1.21) are provided by parameterized module spectra, in particular by flat ∞-vector bundles ("∞-local systems", see [SS23-EoS]) which, in their 0-sector (Rem. 1.22), accommodate quantum circuit semantics (cf. §2.4) in indexed sets of vector spaces (cf. §2.1) such as known from the Proto-Quipper quantum language (Lit. 1.5).

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

Motivic Yoga. LHoTT witnesses these quantum halos as *linear types* (24) equipped with a closed tensor product $\otimes$ and compatible base change operations which satisfy the rules of Grothendieck's "motivic yoga of six operations" in Wirthmüller style (Def. 2.18, cf. [Ri22a, §2.4][SS23-EoS, §3.3]). It is this "motivic" structure from which the structure of quantum physics derives, as originally observed in [Sch14a] and here brought out in §2.1.

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 W : ClaType$^{\text{fin}}$)	$\overset{\text{cart. product}}{\prod\limits_W \mathcal{H}_w} \simeq \underset{\text{direct sum}}{\oplus \mathcal{H}_w} \simeq \overset{\text{co-product}}{\coprod\limits_W \mathcal{H}_w}$	$\mathscr{V} \otimes \big(\bigoplus\limits_{w:W} \mathcal{H}_w \big) \simeq \bigoplus\limits_{w:W} (\mathscr{V} \otimes \mathcal{H}_w)$	$(\mathscr{V} \otimes \mathcal{H}) \multimap \mathscr{K}$ $\simeq \mathscr{V} \multimap (\mathcal{H} \multimap \mathscr{K})$
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

***H*$\mathbb{C}$-Linear quantum theory.** In this scheme, conventional quantum information theory happens in the $\mathbb{C}$-linear form of linear homotopy theory (details in [SS23-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-sector (Rem. 1.22) these are just set-indexed complex vector spaces of the form familiar from the categorical semantics of the quantum language Quipper, this is what we discuss in detail §2.1 . But since all our quantum effects are constructed monadically (§2) relying just on the abstract Motivic Yoga, they apply at once to unrestricted (stable) homotopy types, providing a homotopy-theoretic form of quantum mechanics suitable for the discussion of "topological quantum effects" as in [TQP].

---

[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].

0.2 Quantum Monadology

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. 1.4), *what makes it applicable to certification of quantum effects such as quantum measurement* (Lit. 1.2)?

A universal quantum programming language has to accurately reflect the logical content of quantum physics, where the act of formulating a quantum program is also that of recounting, in formalized language, the physical process of its execution. The execution of quantum programs *includes* processes of quantum measurement and therefore any formulation must handle 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. 1.13), 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):

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

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. 1.14) where we consider any W-dependent data types:[3]

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

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. 1.17, 1.14):

We observe in §2.2 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. 2.21) 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.

$$\begin{tikzcd}[column sep=huge]
\mathrm{Type}_W \arrow[loop above, "\overset{\textbf{possibility modality}}{\Diamond_W}"] \arrow[loop below, "\underset{\textbf{necessity modality}}{\Box_W}"'] \arrow[rr, shift left=4, "\overset{\textbf{dependent “sum”}}{\coprod_W}"] \arrow[rr, shift right=4, "\underset{\textbf{dependent product}}{\prod_W}"'] \arrow[rr, phantom, shift left=2, "\perp"] \arrow[rr, phantom, shift right=2, "\perp"] & & \mathrm{Type} \arrow[ll, "\times W" description] \arrow[loop above, "\overset{\textbf{randomness modality}}{\star_w}"] \arrow[loop below, "\underset{\textbf{indefiniteness modality}}{\bigcirc_w}"']
\end{tikzcd} \tag{4}$$

While for classical intuitionistic type theory, this perspective may be of interest to the analytic philosopher (see [Cor20, 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 like `LHoTT` the situation (4) has an immediate analog ([Ri22a, §2.4]) as W-dependent classical intuitionistic types are replaced by W-dependent *linear* types (quantum data types, interpreted for instance a indexed sets of vector spaces, see §2.1): 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. 1.18) on the *direct sum* of linear types, this reflecting the superposition principle of quantum physics:

linear possibility **linear randomness**

direct sum

Frobenius monad of quantum epistemic logic (§2.3) proves principles as *deferred measurement* (Prop. 2.40)

$$\begin{tikzcd}
W : \mathrm{ClaType}^{\mathrm{fin}} \vdash & \mathrm{QuType}_W \arrow[loop above, "\Diamond_W"] \arrow[loop below, "\Box_W"] \arrow[r, shift left=3, "\oplus_W"] \arrow[r, phantom, shift left=1.5, "\perp"] \arrow[r, phantom, shift right=1.5, "\perp"] \arrow[r, shift right=3, "\oplus_W"'] & \mathrm{QuType} \arrow[l] \arrow[loop above, "\star_W"] \arrow[loop below, "\bigcirc_W"]
\end{tikzcd} \tag{5}$$

Classical context (Prop. 2.35) Frobenius monad as in `zxCalculus` gives *effect-logic* for quantum gates §2.4

direct sum

linear necessity **linear indefiniteness**

This means equivalently that in the linear case the (co)monadic modal operators coincide, $\Diamond_W \simeq \Box_W$, $\star_W \simeq \bigcirc_W$, to form a pair of *Frobenius monads* (cf. Prop. 2.35), reflecting the monadic nature of quantum measurement as known from the `zxCalculus` (Lit. 1.18). 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**

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

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. 2.40) 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 (cf. [AC07]): 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. 1.2):

Classical register **quantum register** **QBit-measurement branching** (pp. 84) **QBit-controlled quantum gate** **dynamic lifting**

$$\begin{tikzcd}
b : \mathrm{Bit} \vdash & \mathcal{H} \arrow[r, "P_0"] \arrow[dr, "P_1"'] & \mathcal{H} \arrow[r, "G"] & \mathcal{K} \arrow[dr, "\delta^b_0"] & \\
& & \oplus \quad \oplus & \oplus & \mathcal{K} \\
& & \mathcal{H} \arrow[r, "G"] & \mathcal{K} \arrow[ur, "\delta^b_1"'] &
\end{tikzcd}$$

The hexagon of quantum epistemic entailments.
A commuting diagram (189) 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 $\mathcal{H}_i = H$).

$$\begin{tikzcd}
& \Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}\mathcal{H}_\bullet \arrow[rr, "\Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}G_\bullet"] & & \Box_{\mathrm{Bit}}\Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet \arrow[dr, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet}"] & \\
\Box_{\mathrm{Bit}}\mathcal{H}_\bullet \arrow[ur, "\Box_{\mathrm{Bit}}\left(\mathrm{ret}^{\Diamond_{\mathrm{Bit}}}_{\mathcal{H}_\bullet}\right)"] \arrow[rrrr, "\text{classically controlled quantum computing cycle}" description] \arrow[dr, "\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathcal{H}_\bullet}"'] & & & & \Diamond_{\mathrm{Bit}}\mathcal{K}_\bullet \\
& \mathcal{H}_\bullet \arrow[rr, "G_\bullet"] & & \mathcal{K}_\bullet \arrow[ur, "\mathrm{ret}^{\Diamond_{\mathrm{Bit}}}_{\mathcal{K}_\bullet}"'] &
\end{tikzcd} \tag{7}$$

quantum effects Everett-style

quantum effects Copenhagen-style

$$\begin{tikzcd}
b : \mathrm{Bit} \vdash & \mathcal{H} \arrow[r] & \mathcal{H}_b \arrow[r, "G_b"] & \mathcal{K}_b \arrow[r, hook] & \mathcal{K} \\
& \sum\limits_{b' \in \mathrm{Bit}} \vert \psi_{b'} \rangle \arrow[r, mapsto] & \vert \psi_b \rangle \arrow[r, mapsto] & G_b \vert \psi_b \rangle &
\end{tikzcd}$$

QBit-measurement collapse (pp. 81) **quantum gate conditioned on classical control logic** **dynamic QBit-state preparation**

Monads as computational effects. In a curious generalization of modal logic to functional programming (Lit. 1.16), monads on a category of data types serve to encode *computational effects* (Lit. 1.17). For instance, a classical program whose output data type is *nominally* D but *de facto* the value $\bigcirc_W D$ of the classical W-*indefiniteness monad* (4) — often known as the *Reader*- or *Environment*-monad (79) — actually produces its D-valued output only conditioned on the observation ("reading") of an indefinite variable ("environment" state) $w : W$, hence on a classical W-*measurement*, so to speak. In this sense, a program of the type $D \to \bigcirc_W D'$ has a classical *measurement effect* – quite literally: in its generalized incarnation as the IO-monad (83) in `Haskell`, running such a procedure causes the computer to perform a read-out of its RAM-state (86):

plain input data **indefiniteness-effectful output data** (67) **effect-bound (Kleisli) composition** (69)

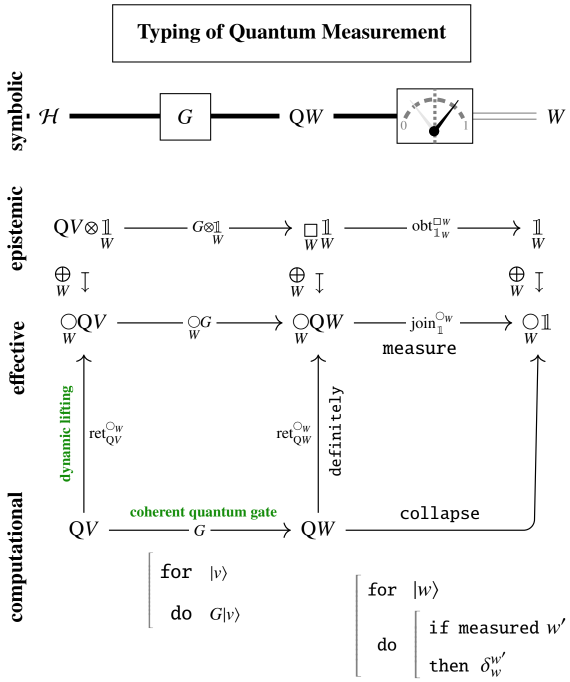
$$\begin{array}{ccc} f_\bullet : D \xrightarrow[\text{program}]{\text{effectful}} \underset{W}{\bigcirc} D' & g_\bullet : D_2 \xrightarrow[\text{program}]{\text{effectful}} \underset{W}{\bigcirc} D_3 & g_\bullet \mathrel{>=>} f_\bullet : D \longrightarrow \underset{W}{\bigcirc} D' \\ d \quad \mapsto \quad (w \mapsto f_w(d)) & d \quad \mapsto \quad (w \mapsto g_w(d)) & d \quad \mapsto \quad (w \mapsto g_w \circ f_w(d)) \end{array} \tag{8}$$

Quantum measurement as computational effect. Now, in contrast to classical computing, in the quantum case the right adjoint $\oplus_W$ in (5) is a *monadic functor* (Prop. 2.32), meaning that the W-dependent quantum types are equivalently the *modal types* (93) — also called *modules*, but we will say *modales* for brevity and for emphasis of the modal perspective – over the *quantum indefiniteness* monad $\bigcirc_W$ appearing on the other side of this ambidextrous adjunction (5).

Under this equivalence, the $\square_W$-obtain operation which gives quantum state collapse in (7) is now reflected in the $\bigcirc_W$-join operation constituting a *computationally effective* typing of the previously *epistemic* typing of quantum measurement (see §2.4, p. 85 and §3.1, p. 104).

The (co)monadic formalization of quantum measurement in the `zxCalculus` (Lit. 1.18) derives from this formulation (cf. Prop. 2.35, Rem. 2.42).

But by understanding this monad as a computational effect, we may apply a general method for articulating monadic effects in programming language (`do`-notation, Lit. 1.19) to obtain a natural *Quantum Systems*-language (`QS`, §3, a domain-specific language embeddable into `LHoTT`) naturally coding parameterized quantum circuits with measurement effects.



Mixed quantum measurement as monoidal-monadic effect. The quantum indefiniteness-monad $\bigcirc_W$ is in fact a *strong monad* (Prop. 2.37). Besides guaranteeing (77) that it really does exist as a programming language construct, this means that it carries a symmetric monoidal monad structure (78) $\mathrm{pair}^{\bigcirc_W}$ (210). We observe (221) that this monoidal monad structure serves to enhance the above computational typing of measurement effects from pure to mixed quantum states (35), where it embodies the *Born rule* (32) of quantum measurement in its form originally identified by Lüders (44):

mixed states **density matrices** **measure separately states and co-states** **decohere: discard off-diagonal entries** **probability distributions**

$$\begin{tikzcd}[column sep=large] \begin{array}{c}\mathrm{Q}W \\ \otimes \\ (\mathrm{Q}W)^*\end{array} \arrow[r, "\simeq"] \arrow[rrrr, bend right, "\texttt{collapse}_W"'] & \begin{array}{c}\oplus_W \mathbb{C} \\ \otimes \\ \oplus_W \mathbb{C}^*\end{array} \arrow[r, "\begin{array}{c}\texttt{collapse}_w \equiv \mathrm{join}^{\bigcirc_W}_{\mathbb{C}} \circ \mathrm{ret}^{\bigcirc_W}_{\oplus_W \mathbb{C}} \\ \otimes \\ \texttt{collapse}_w \equiv \mathrm{join}^{\bigcirc_W}_{\mathbb{C}^*} \circ \mathrm{ret}^{\bigcirc_W}_{\oplus_W \mathbb{C}^*}\end{array}"] & \begin{array}{c}\bigcirc_W \mathbb{C} \\ \otimes \\ \bigcirc_W \mathbb{C}^*\end{array} \arrow[r, "\mathrm{pair}^{\bigcirc_W}_{\mathbb{C},\mathbb{C}^*}"] & \bigcirc_W \begin{array}{c}\mathbb{C} \\ \otimes \\ \mathbb{C}^*\end{array} \arrow[r, "\bigcirc_W \mathrm{ev}"] & \bigcirc_W \mathbb{C} \end{tikzcd} \tag{9}$$

tensor product of free $\bigcirc_W$-modales separately handle pure $\bigcirc_W$- effects **monoidal monad structure on $\bigcirc_W$**

$$|\psi\rangle\langle\psi| \;\mapsto\; \begin{array}{c}\big(\textstyle\sum_w |w\rangle\langle w|\psi\rangle\big) \\ \otimes \\ \big(\textstyle\sum_{w'} \langle\psi|w'\rangle\langle w'|\big)\end{array} \;\mapsto\; \Big((w,w') \mapsto |w\rangle \underbrace{\langle w|\psi\rangle\langle\psi|w'\rangle}_{\text{coherences}} \langle w'|\Big) \;\mapsto\; \big(w \mapsto \big|\langle w|\psi\rangle\big|^2\big)$$

a pure state among mixed **Born rule**

Moreover, postcomposition with the monoidal monad structure $\mathrm{pair}^{\bigcirc_W}$ makes the enhancement of parameterized quantum circuits from pure to mixed states a functor of $\bigcirc_W$-effectful maps (214),

$$\begin{tikzcd}
\mathrm{QuType}_{\bigcirc_W} \arrow[rrrr, "\text{enhancement to mixed states}"] & & & & \mathrm{QuType}_{\bigcirc_W} & \\
\mathcal{H}_1 \arrow[r, "G_\bullet"] & \underset{W}{\bigcirc}\mathcal{H}_2 & \longmapsto & \begin{array}{c}\mathcal{H}_1\\ \otimes\\ \mathcal{H}_1^*\end{array} \arrow[r, "\begin{array}{c}G_\bullet\\ \otimes\\ G^{\dagger *}_\bullet\end{array}"] \arrow[rr, bend right, "(G\otimes G^{\dagger *})_\bullet"'] & \begin{array}{c}\underset{W}{\bigcirc}\mathcal{H}_2\\ \otimes\\ \underset{W}{\bigcirc}\mathcal{H}_2^*\end{array} \arrow[r, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_1,\mathcal{H}_1^*}"] & \underset{W}{\bigcirc}\begin{array}{c}\mathcal{H}_2\\ \otimes\\ \mathcal{H}_2^*\end{array}
\end{tikzcd} \tag{10}$$

in that it respects (Lem. 2.39) their effect-bound (Kleisli) composition (8):

$$(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_2,\,\mathcal{H}_2^*} \circ (G_\bullet \otimes G_\bullet^{\dagger *})) >=> (\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\,\mathcal{H}_3^*} \circ (H_\bullet \otimes H_\bullet^{\dagger *})) \quad = \quad \mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\,\mathcal{H}_3^*} \circ \big((G_\bullet >=> H_\bullet) \otimes (G_\bullet^{\dagger *} >=> H_\bullet^{\dagger *})\big). \tag{11}$$

This means that the above computational effective typing of parameterized quantum circuits with quantum measurement enhances *verbatim* from pure to mixed states!

The modal quantum logic QuantumState. We go one step further and observe (§2.5) a *modal-logical origin* even of the notion of mixed quantum states (35) and the quantum channel operations between them. Namely, observing

$$\underset{W}{\text{☆}}\mathcal{K} \;\equiv\; \underset{W}{\bigoplus}\mathcal{K} \;\simeq\; \underset{W}{\bigoplus}(\mathcal{K}\otimes\mathbb{1}) \;\simeq\; \mathcal{K}\otimes\big(\underset{W}{\bigoplus}\mathbb{1}\big) \;\equiv\; \mathcal{K}\otimes\mathrm{Q}W$$

density matrices are identified among the "indefinitely random scalars":

$$\textbf{Q}\boldsymbol{W}\textbf{-(density-)matrices} \quad \mathrm{Q}W\otimes\mathrm{Q}W^* \quad\simeq\quad \underset{W}{\bigcirc}\underset{W}{\text{☆}}\mathbb{1} \quad \begin{array}{c}\boldsymbol{W}\textbf{-indefinitely }\boldsymbol{W}\textbf{-random}\\ \textbf{scalars}\end{array}$$

This equivalence ranges deeper – it is actually an equivalence of the corresponding monads, and as such eventually is the modal-logical reason for *unitarity* of quantum gates – as follows:

Generally, for *dualizable* (133) – namely finite-dimensional – quantum types $\mathcal{H} \,:\, \mathrm{QuType}^{\mathrm{fdm}}$ their tensoring-functors again are in ambidextrous adjunction (135), yielding another Frobenius monad (cf. Rem. 2.44) — the linear/quantum version of the classical State-monad (83):

$$\begin{tikzcd}
\textbf{QuantumState} & \textbf{QuantumStore} & & \textbf{QuantumState} & \textbf{QuantumStore} \\
\mathrm{QuType} \arrow[loop above, "\mathrm{Q}W\mathrm{State}"] \arrow[loop below, "\mathrm{Q}W^*\mathrm{Store}"] \arrow[r, shift left=4, "\text{☆}_W"] \arrow[r, shift left=2, phantom, "\perp"] \arrow[r, shift right=2, phantom, "\perp"] \arrow[r, shift right=4, "\text{☆}_W"'] & \mathrm{QuType} \arrow[l, "\bigcirc_W" description] \arrow[loop above, "\mathrm{Q}W^*\mathrm{Store}"] \arrow[loop below, "\mathrm{Q}W\mathrm{State}"] & \text{generally:} & \mathrm{QuType} \arrow[loop above, "\mathcal{H}\mathrm{State}"] \arrow[loop below, "\mathcal{H}^*\mathrm{Store}"] \arrow[r, shift left=4, "(-)\otimes\mathcal{H}"] \arrow[r, shift left=2, phantom, "\perp"] \arrow[r, shift right=2, phantom, "\perp"] \arrow[r, shift right=4, "(-)\otimes\mathcal{H}"'] & \mathrm{QuType} \arrow[l, "(-)\otimes\mathcal{H}^*" description] \arrow[loop above, "\mathcal{H}\mathrm{Store}"] \arrow[loop below, "\mathcal{H}^*\mathrm{State}"]
\end{tikzcd} \tag{12}$$

This identifies the $\mathrm{Q}W$State-monad with the monad that is induced, in turn, by the epistemic indefiniteness/randomness adjunction $\bigcirc_W \dashv \text{☆}_W$ (5):

$$\textbf{QuantumState} \quad \mathrm{Q}W\mathrm{State} \;\equiv\; \mathrm{Q}W \multimap ((-)\otimes\mathrm{Q}W) \;\simeq\; (-)\otimes\mathrm{Q}W\otimes\mathrm{Q}W^* \;\simeq\; \underset{W}{\bigcirc}\underset{W}{\text{☆}} \quad \begin{array}{c}\textbf{Quantum}\\ \textbf{indefinite}\\ \textbf{randomness}\end{array}$$

By itself, the QuantumState monad encodes qRAM-effects (217), in quantization of the RAM-effect (86) of classical State-monads. But with its *monad transformations* (103) taken into account it models quantum channels (39):

Distributing Frobenius monads at the heart of quantum information theory. The QuantumState (co)monads pairwise distribute over the QuantumEnvironment (co)monads (Prop. 2.56), which implies

(i) 2-sided Kleisli categories (126) of (Prop. 2.58):

  (a) QuantumEnvironment-contextful & QuantumState-effectful maps modelling mixed state preparation, eg. $\text{☆}_W\mathbb{1} \to \mathcal{H}\otimes\mathcal{H}^*$

  (b) QuantumState-effectful & QuantumEnvironment-contextful maps modelling mixed state observables, eg. $\mathcal{H}\otimes\mathcal{H}^* \to \bigcirc_W\mathbb{1}$

acted on by QuantumState- and QuantumStore-transformations, respectively.

(ii) the composite monads $\bigcirc_W \circ \mathcal{H}\mathrm{State} \;\dashv\; \mathcal{H}\mathrm{Store}\circ\text{☆}_W$ exist (114).

$$\begin{array}{ccccc}
\begin{array}{c}\textbf{Quantum}\\ \textbf{indefiniteness}\end{array} & & \begin{array}{c}\textbf{Quantum}\\ \textbf{environment}\end{array} & & \begin{array}{c}\textbf{quantum}\\ \textbf{randomness}\end{array} \\
\underset{W}{\bigcirc} & \simeq & (-)\otimes\mathrm{Q}W & \simeq & \underset{W}{\text{☆}} \\
\underset{W}{\bigcirc}\underset{W}{\text{☆}} & \simeq & (-)\otimes\,\mathrm{Q}W\otimes\mathrm{Q}W^* & \simeq & \underset{W}{\text{☆}}\underset{W}{\bigcirc} \\
\begin{array}{c}\textbf{Quantum}\\ \textbf{indefinite}\\ \textbf{randomness}\end{array} & & \begin{array}{c}\textbf{quantum}\\ \textbf{state}\end{array} & & \begin{array}{c}\textbf{Quantum}\\ \textbf{random}\\ \textbf{indefiniteness}\end{array}
\end{array}$$

$$\mathrm{Monads} \longleftarrow \mathrm{FrobMonads} \longrightarrow \mathrm{CoMonads}$$

Unitary quantum channels are QuantumState-transformations. In fact, the composition of QuantumState monads with the indefiniteness-modality is *itself* a relative monad *on the category of QuantumState monads* (Prop. 2.69):

$$\begin{tikzcd}
\text{Indefiniteness of pure states} & \underset{W}{\bigcirc} : & \mathrm{QuType} \arrow[rr] & & \mathrm{QuType} & \text{Monad on quantum types} \\
& & \mathcal{H} \arrow[rr, mapsto] & & \underset{W}{\bigcirc}\mathcal{H} & \\
\text{Indefiniteness of mixed states} & \underset{W}{\bigcirc} \circ : & \mathrm{StateMnd}(\mathrm{QuType}) \arrow[rr] & & \mathrm{Mnd}(\mathrm{QuType}) & \text{Relative monad on QuantumState monads} \\
& & \mathcal{H}\mathrm{State} \arrow[rr, mapsto] & & \underset{W}{\bigcirc} \circ \mathcal{H}\mathrm{State} &
\end{tikzcd} \tag{13}$$

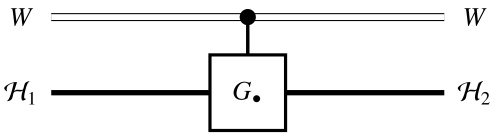
This is such that the enhancement (10) of indefiniteness-effectful maps from pure to mixed states is a QuantumState transformation iff the maps are unitary, W-wise (Prop. 2.70):

Where pure quantum states are terms of linear (quantum) type $\mathcal{H}$ (24), the (ambient, linear) type of *mixed states* in the form of (density) matrices may be identified with the QuantumState-*monad* $\mathcal{H}$State (12) acting on these linear types: Where a quantum circuit of pure states is a map of linear (quantum) types, a quantum circuit of mixed states is a *transformation of monads* (103) of QantumState monads – a *QuantumState transformation*.

It is with this natural typing of quantum circuits literally as QuantumState transformations that the unitarity axiom of quantum physics is reflected in modal quantum logic.

Moreover, the indefiniteness-modality $\bigcirc_W$ on quantum types enhances to a (relative) *monad on QuantumState monads* (Prop. 2.69), such that the $\bigcirc$-modal typing of parameterized quantum circuits (§2.4) is formally the same for pure and mixed states, under the enhancement $\mathcal{H} \mapsto \mathcal{H}$State of underlying categories of types from QuType to StateMnd(QuType).

Parameterized quantum circuit



$$\begin{tikzcd}
\text{Typing for pure states} & \underset{W}{\bigcirc}\mathcal{H}_1 \arrow[rr, "G_\bullet", "\text{indefinite linear map}"'] \arrow[d, phantom, "\top"] & & \underset{W}{\bigcirc}\mathcal{H}_2 \\
& \text{enhanced via (10) iff all } G_w \text{ are unitary} \arrow[d] & & \\
\text{Typing for mixed states} & \underset{W}{\bigcirc} \circ \mathcal{H}_1\mathrm{State} \arrow[rr, "(G \otimes G^{\dagger *})_\bullet", "\text{indefinite QuantumState monad transformation}"'] & & \underset{W}{\bigcirc} \circ \mathcal{H}_2\mathrm{State} \\
& \underset{W}{\bigoplus} \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[rr, "\text{is in components parameterized}", "\text{unitary quantum channel}"'] & & \underset{W}{\bigoplus} \mathcal{H}_2 \otimes \mathcal{H}_2^* \\
& (w,\, \rho) \arrow[rr, mapsto] & & (w,\, G_w \cdot \rho \cdot G_w^\dagger)
\end{tikzcd} \tag{14}$$

These unitary quantum channels are also QuantumStore-comonad transformations, and as such their action (111) on the quantum observables typed as QuantumStore-contextful scalars (Ex. 2.46) gives *Heisenberg evolution* (Prop. 2.50):

$$\begin{tikzcd}
\mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "O_A", "\text{Observable = QuantumState-contextful scalar}"'{pos=0.3, yshift=8ex}] & \mathbb{1} \arrow[r, phantom, "\overset{\text{acted on by unitary QuantumStore transformation}}{\mapsto}"] & \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "U \otimes U^{\dagger *}"] \arrow[rr, bend left, "O_{U^\dagger \cdot A \cdot U}"] & \mathcal{H}_2 \otimes \mathcal{H}_2^* \arrow[r, "O_A"] & \mathbb{1} \\
\rho \arrow[r, mapsto] & \mathrm{Tr}(\rho \cdot A) & \rho \arrow[r, mapsto] & U \cdot \rho \cdot U^\dagger \arrow[r, mapsto] & \mathrm{Tr}(\rho \cdot U^\dagger \cdot A \cdot U)
\end{tikzcd} \tag{15}$$

General quantum channels. The other canonical example of a QuantumState-monad transformation is the (quantum channel given by) coupling (tensoring) to a uniform bath state (56), whose formal dual is the QuantumStore-comonad transformation given by partial trace

$$\mathrm{couple}^{\mathcal{H}} \;:\; \mathcal{H}\mathrm{State} \longrightarrow (\mathcal{H} \otimes \mathcal{B})\mathrm{State} \qquad \mathrm{average}^{\mathcal{B}} \;:\; (\mathcal{H} \otimes \mathcal{B})\mathrm{Store} \longrightarrow \mathcal{H}\mathrm{Store}\,.$$

This way, every *unistochastic quantum channel* (53) appears as a composite of a QuantumState transformation followed by a QuantumStore-transformation, and as such acts (106) on the 2-sided Kleisli categories (Lem. 2.60) of quantum observables and quantum state preparations.

As a simple but relevant example, the DQC1-model of quantum computation (54) on a single ("clean") qbit coupled to a uniformly distributed bath is naturally typed in this monadic language as follows:[4]

$$\begin{tikzcd}
\text{(Ex. 2.59)} & \star_{\mathrm{Bit}}\mathbb{1} \arrow[d, "\mathrm{weigh}_{\delta^0_\bullet} >=> \mathrm{prep}"'] \arrow[dr, bend left, "\text{QuantumState effect}"] \arrow[dr, "\textbf{preparation}"'] & & & \\
& \mathrm{QBitState}(\mathbb{1}) \arrow[r, "\mathrm{couple}^{\mathcal{B}}"'] \arrow[rr, bend right, "\text{QuantumState transformation}"'] & (\mathrm{QBit} \otimes \mathcal{B})\mathrm{State}(\mathbb{1}) \arrow[r, "\mathrm{chan}^{U_{\mathrm{tot}}}", "\textbf{evolution}"'] \arrow[rr, bend left, "\text{QuantumStore transformation}"] & (\mathrm{QBit} \otimes \mathcal{B})\mathrm{State}(\mathbb{1}) \arrow[r, "\mathrm{average}^{\mathcal{B}}"] \arrow[dr, "\textbf{measurement}"] \arrow[dr, bend right, "\text{QuantumStore context}"'] & \mathrm{QBitState}(\mathbb{1}) \arrow[d, "\mathcal{O}_{P_\bullet}"] \\
& & & & \bigcirc_{\mathrm{Bit}}\mathbb{1}
\end{tikzcd} \tag{16}$$

$$\begin{tikzcd}
|0\rangle\langle 0| \arrow[r, mapsto] & |0\rangle\langle 0| \otimes I_{\mathcal{B}} \arrow[r, mapsto] & U_{\mathrm{tot}}(|0\rangle\langle 0| \otimes I_{\mathcal{B}})U_{\mathrm{tot}} \arrow[dr, mapsto] & \\
& & & b \mapsto \mathrm{Tr}^{\mathrm{QBit}}\Big(|b\rangle\langle b|\, \mathrm{Tr}^{\mathcal{B}}\big(U_{\mathrm{tot}}(|0\rangle\langle 0| \otimes I_{\mathcal{B}})U_{\mathrm{tot}}\Big) \arrow[d, equal] \\
& & & b \mapsto \mathrm{Tr}\Big((|b\rangle\langle b| \otimes I_{\mathcal{B}})\, U_{\mathrm{tot}}(|0\rangle\langle 0| \otimes I_{\mathcal{B}})U_{\mathrm{tot}}\Big)
\end{tikzcd}$$

Monadic typing of DQC1 quantum channel (54). The quantum channel as such is the horizontal composite, consisting of an environmental coupling, followed by joint unitary evolution, followed by bath-averaging (and then by quantum measurement in the form (9)). Here the first two steps constitute a QuantumState transformation acting compatibly (105) on the QuantumState-effectful maps which prepare the state (on the left). Dually, the last two steps constitute a QuantumStore-transformation which acts compatibly on the QuantumStore-contextful observables (by Heisenberg evolution, cf. Prop. 2.50) to produce the measurement result, as in (15).

Effective quantum language from Quantum modal logic. With this thoroughly modal/monadic formulation of quantum systems in hand, standard language constructs in functional programming for handling effect monads (Lit. 1.19) become available for quantum programming. We indicate the resulting *Quantum Systems Language* (`QS`) in §3.

Outlook. While one motivation for all these monadic constructions is the remarkable fact that they can be embedded just by suitable sugaring (Lit. 1.6) into any dependent linear type theory which verifies the Motivic Yoga (such as `LHoTT` does, Lit. 1.8), here we speak purely in categorical semantics and relegate all discussion of type theoretic syntax to elsewhere (but for a preview of the translation see [Ri23]). At the same time, `LHoTT` exists for the moment only on paper, as it is not supported yet by the `HoTT` proof assistants such as `Agda` or `Coq`. There should be no fundamental obstacle to implementing a linear version of, say, `Agda`, but this will require dedicated work. Therefore we understand our contribution here also as demonstrating that the new type system `LHoTT` (which might superficially seem to be of only specialized interest) fundamentally deserves the attention of the computer-proof-assistant community.

Similarly, here we do not dwell on the higher homotopy theoretic aspect of `LHoTT`/`QS`; but the companion article [TQP] discusses in detail how anyonic topological quantum gates are naturally realized in classical `LHoTT`, namely as twisted higher cohomology groups realized as dependent function types into higher delooping types (Eilenberg-MacLane-spaces) of the type of complex numbers. Since `LHoTT` is conservative over `HoTT`, this same construction from [TQP] may immediately be understood as taking place in `LHoTT`, where the type of complex numbers and hence that of anyonic quantum ground states may now be promoted to genuine linear types (Eilenberg-MacLane spectra equivalent to chain complexes, via the categorical semantics in [SS23-EoS]), exhibiting the actual Hilbert space type of anyons to which quantum circuit logic may then be applied in the way we are discussing here.

Efforts are underway at CQTS[5] to implement this classical `HoTT`-realization of topological quantum gates in cubical-`Agda` in order to demonstrate the feasibility of a formally verified topological hardware-aware quantum programming/simulation environment via dependent type theory. Our aim here is to demonstrate that the linearly-typed enhancement of such a quantum language system is theoretically viable, and naturally so, hoping to thereby spur its eventual implementation.

Acknowledgements. We thank Thorsten Altenkirch, Nathanael Arkor, David Corfield, David Jaz Myers, Mitchell Riley, and Sachin Valera for useful discussion concerning various aspects of this paper.

---

[4] Notice that the environmental mixed state produced by this construction is un-normalized. This is no restriction of generality, it just means that for extracting actual probabilities one needs to normalize by the trace of the density matrix.

[5] landing page: `nyuad.nyu.edu/en/research/faculty-labs-and-projects/cqts.html`

1 Background

This section provides background information and pointers to the literature on the 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 any remaining gaps that we address in the main text.

1.1 Quantum computing

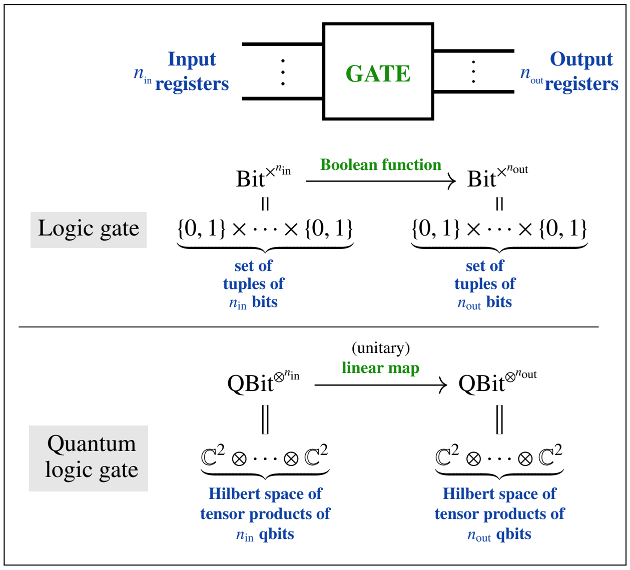
Literature 1.1 (**Quantum computation and Quantum information processing).**
The basic idea of *quantum computation* and *quantum information processing* is to exploit, for the purpose of machine computation and information processing, the peculiar laws of quantum physics (Lit. 1.2) – which are obeyed by *undisturbed* (Lit. 1.3) 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 can be technologically retained (cf. Lit. 1.3), which has so far been achieved only marginally (Lit. 1.10).

Textbook accounts of the general principles of quantum computation and quantum information theory include: [NC00] [RP11][BCR18][BEZ20], lecture notes include [Pre04]. Impressions of the state of the field may be found in [Pr22]. An exposition leading up to our discussion here may be found in [Sch22].

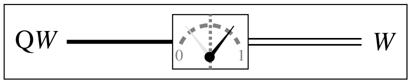
As usual, we are primarily concerned here with "digital" (or "discrete variable") quantum information/computa-tion, where all quantum state spaces are *finite-dimensional*, cf. (133). While there are notions of quantum computation on (separably) infinite-dimensional Hilbert spaces ("continuous variable" systems, e.g. [Cho22]) these represent "analog quantum computation" [KNM10] which, just as its classical analog, is typically more specialized, less reliable and less amenable to theory than "digital" computation on finite (dimensional) state spaces.

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. 1.1) 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" (166) – cf. (64).
In classically controlled quantum computation (Lit. 1.11) one is dealing with *classically controlled quantum gates* (e.g. [NC00, §4.3]) that read/write a combination of classical and quantum data.



An example of a (controlled, quantum) logic gate is the *controlled NOT gate* [De89, Fig. 2] (`CNOT` for short, cf. [NC00, §1.3.2]) which operates on a pair of (q)bits by inverting the second conditioned on the first; see (17) and (254).

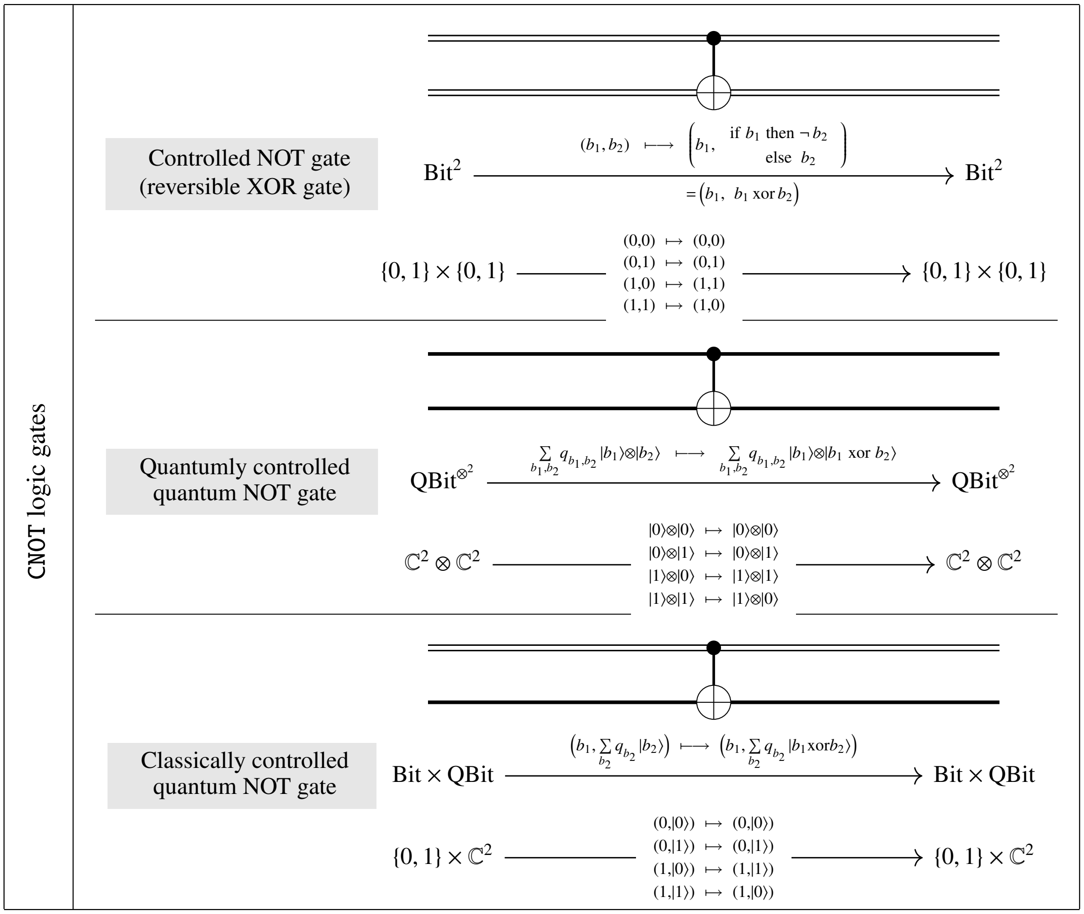
Quantum measurement gates. One also wants to regard the operation of *quantum measurement* itself (Lit. 1.2) as a quantum gate (e.g. [NC00, p. xxv]), whose input is quantum data but whose output is the classical measurement result.



Notice that the proper data-typing (Lit. 1.4) 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]).

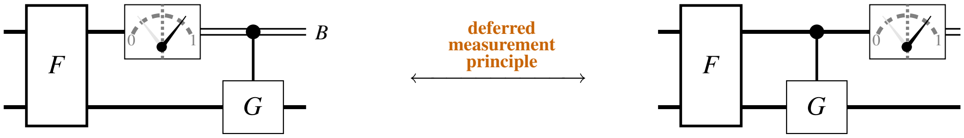
Beware that this is not a side issue but part of the crux of quantum computation: On the one hand, the stochastic nature of quantum measurement is a *fundamental* principle of physics (certainly of presently accessible physics, see Lit. 1.2) and not just a reflection of incomplete knowledge about a quantum 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 §3.2.2 and quantum error correction §3.2.3), 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 §2.4.



(17)

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



(18)

(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. 1.2, can always be thought of as postponed indefinitely.)

The theoretical status of the deferred measurement principle had remained somewhat inconclusive. Available textbooks (e.g. [NC00, §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. 2.40) that the deferred measurement principle (18) is verified in the data-typing of quantum processes provided in `LHoTT` (Lit. 1.8).

Notice that the content of this *equivalence between intermediate and deferred measurement collapse* (18) 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*" and of "Wigner's friend" (where the cat/friend plays the role of the intermediate controlled quantum gate). Moreover, the same paradox, in different words, was influentially offered in [Ev57a, p. 4] as the main argument against the "Copenhagen interpretation" and for the "many-worlds interpretation" of quantum physics (cf. Lit. 1.2). Note that our same formalism which proves (18) 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", also known as a *Mealy machine*, see (86) below):

$$\begin{array}{c}\textbf{read-in RAM}\\ \textbf{\& input data}\end{array}\; \mathrm{RAM}\times D \xrightarrow{\begin{array}{c}\textbf{program interacting with}\\ \textbf{Random Access Memory}\end{array}} \mathrm{RAM}\times D' \;\begin{array}{c}\textbf{write RAM}\\ \textbf{\& output data}\end{array} \tag{19}$$

Starting with [GLM08a][GLM08b], authors envisioned that quantum computing should similarly support a "qRAM model" (see [Liu$^+$23, p. 18] for implementations) the basic idea being that data in qRAM may form quantum superpositions and may coherently be read/written in this form. As with the deferred measurement principle above, existing literature discusses this concept not in general abstraction but by way of concrete examples (see for instance [Ar$^+$15, Fig. 9][PPR19, Fig. 1][PCG23, Fig. 4][6]). 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]):

$$\begin{array}{c}\textbf{read-in qRAM}\\ \textbf{entangled with}\\ \textbf{input quantum data}\end{array}\; \mathrm{QRAM}\otimes\mathcal{H} \xrightarrow{\begin{array}{c}\textbf{quantum program}\\ \textbf{interacting with qRAM}\end{array}} \mathrm{QRAM}\otimes\mathcal{H}' \;\begin{array}{c}\textbf{write qRAM}\\ \textbf{entangled with}\\ \textbf{output quantum data}\end{array} \tag{20}$$

In §2.4 we obtain (217) a formalized account/typing of qRAM and its equivalence to controlled quantum circuits.

Literature 1.2 (Epistemology of quantum physics and its formalization). The curious epistemology[7] of quantum physics ([Di30][vN32], see e.g. [SN94][Ish95][La17]) 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* (see Lit. 1.12) 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 ([vN32, §III.3, §VI][Lü51], cf. [Sche73, §IV][Om94, p. 82][Re22, (A.2)]):

$$\begin{tikzcd}[row sep=small]
 & & & & \mathcal{H}_{w_1} \;\begin{array}{c}\textbf{space of quantum states}\\ \textbf{with definite property } w_1\end{array} \\
\begin{array}{c}\textbf{Hilbert space of all}\\ \textbf{quantum states}\\ \textbf{of the given system}\end{array}\; \mathcal{H} \arrow[r, phantom, "\simeq"] & \Box_W \mathcal{H}_\bullet \arrow[r, phantom, "\equiv"] & \underset{\begin{array}{c}\textbf{direct sum decomposition}\\ \textbf{in measurement basis } W\end{array}}{\underset{w':W}{\bigoplus} \mathcal{H}_{w'}} \arrow[urr, "\text{linear projection}"] \arrow[drr, "\text{linear projection}"'] & & \vdots \\
 & & & & \mathcal{H}_{w_n} \;\begin{array}{c}\textbf{space of quantum states}\\ \textbf{with definite property } w_n\end{array}
\end{tikzcd} \tag{21}$$

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. 1.1 and §2.4, such as in

---

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

[7] Here "epistemology" – the *theory of knowledge* – 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.

the *quantum teleportation* protocol recalled in §3.2.2); 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 (21) *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][Borg08]), whence practicing physicists tend to just disregard the epistemological issue, an attitude that became proverbial under the catch-phrase "shut up and calculate" [Mer89].
Among the main attitudes of quantum philosophers towards the issues are:

- **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 §2, where we present a *modal quantum logic* (cf. Lit. 1.13) 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, p. 4] to dismiss the Copenhagen interpretation and to motivate the "many-worlds" interpretation is arguably resolved by the *deferred measurement principle* (18), which becomes *provable* in quantum modal logic (Prop. 2.40).

Many possible worlds. Previously, several authors (e.g. [Bu76][Sk76, §III][Ta00, p. 101][No02, p. 22][Gi03, §8][Ter19] [Wi20][AA22]) have wondered about or suggested a relation between these "many worlds" of quantum epistemology and the "possible worlds" in the sense classical modal logic (Lit. 1.13) 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. 1.4). (Beware that philosophers also speak of a *modal interpretation of quantum mechanics*[8] 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 this may explain why we have more to say here about the foundations of quantum physics generally, beyond the field of 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. 1.4). 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 that 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 *Linear Homotopy Type Theory* its name. We relate this to algebras of observables in §2.5 (see Ex. 2.46).

Literature 1.3 (**Topological quantum computation).**

---

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

(For extensive motivation, explanation and referencing of topological quantum computation see the companion article [TQP].) The practical promise of quantum computation (Lit. 1.1) hinges on the achievability of fairly *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. §3.2.3).
- **(ii) Topological error protection**: Operate on intrinsically stable quantum hardware (Lit. 1.23) 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. 1.4) until we argued, in the companion article [TQP], that realistic topological quantum gates are naturally modeled by *homotopy typed languages* (Lit. 1.7), such as classical `HoTT` and, more accurately, by `LHoTT` (Lit. 1.8).

Literature 1.4 (**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][9][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. 1.2), 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":

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 (find references around [TQP, (92)]).

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][Luo94][Gu95][Con11][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 (22) from the "space" of data of its input type to that of its output type — called its *denotational semantics* (a seminal idea due to [Sc70][ScSt71]; for exposition see [SK95, §9]):

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

(22)

---

[9] [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.*"

For classical[10] 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 inference rules" called the *contraction rule* and the *weakening rule* ([Ge35, §1.2.1], see [Ja94][Ja98, p. 122][UFP13, §A.2.2][Rij18, §1.4]), which semantically express the possibility of duplicating and of discarding classical data:

	Syntax	**Semantics**
structural inference rules for classical data types	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)

(23)

The quest for quantum data typing was historically convoluted (starting with the much debated quantum logic of [BvN36] and continuing with the influential ideosyncracies of [Gir87]) but is, in hindsight, fairly straightforward: Since the hallmark of coherent quantum evolution is (see [Aby09] for a structural account) the pair of:

- the *no-cloning theorem* ([WZ82], 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 (23) for the logical conjunction of quantum data, which is then called the (non-Cartesian) *tensor product* ⊗ (Lit. 1.20). 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* (see e.g. [BZ06]).

Such *sub-structural* languages were essentially introduced in (the "multiplicative sector" of) the *linear logic* (see [Se89] [Tr92][MN13]) originated by [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. 1.5) 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 (23), which implies that functions $f : X \to Y$ between linear types must indeed use their argument $x : X$ linearly, in the algebraic sense.

Vector- and Hilbert-spaces as linear types. Notably the usual categories $\mathrm{Mod}_{\mathbb{K}}$ of vector spaces over any ground field $\mathbb{K}$, with $\mathbb{K}$-linear maps between them, constitute categorical semantics for (the multiplicative sector) of linear logic, arguably the natural such semantics [1]:

Linear logic is best seen as the realization of the Curry-Howard isomorphism for linear algebra.

The fact that this was made explicit no earlier than in [Mur14][VZ14] must be understood as solely reflecting the convoluted history of the subject: Constituting the heart (cf. Rem. 1.22) of stable ∞-categories of module spectra ($H\mathbb{K}$-modules, in this case, Lit. 1.21) these categories $\mathrm{Mod}_{\mathbb{K}}$ appear as rather canonical models for linear types and as such we use them in §2.1.

Quantum data typing. In summary, the match between quantum phenomena, linear type theories and their semantics in categories of linear spaces is tight (which should not be surprising in hindsight but was less than obvious for much of the history of linear logic):

Quantum Phenomena	**Linear Type Inference**	**Linear maps in Linear algebra...**
No-cloning theorem	Absence of contraction rule	...use their argument at most once.
No-deleting theorem	Absence of weakening rule	...use their argument at least once.

(24)

---

[10] Here by *classical types* we mean the types of *intuitionistic* Martin-Löf type theory in contrast to *linear* (quantum) types (24), 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 (23) 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 "intuitionistic logic").

The resulting principle that

Quantum data has linear type.

has meanwhile come to be more commonly appreciated (e.g. [DLF12, p. 1]) in particular in quantum language design (Lit. 1.5, cf. in particular [FKS20]), where for instance the insightful [Sta15] states up front that:

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 (×) being accompanied by a linear multiplicative conjunction (⊗), 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:

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

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][Ri22a, p. 19]. The claim of the type theory introduced in [Ri22a] 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`[11] (Lit. 1.8).

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. 1.2), 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 the 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. 1.13) 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. 2.10 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. [Mel09, 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. 1.17) and specifically as a comonad induced by a suitably monoidal adjunction (74) between linear and classical (intuitionistic) types (due to [Bi94, p. 157][Be95]):

$$\begin{tikzcd}
\underset{\substack{\textbf{purely}\\ \textbf{classical}\\ \textbf{(intuitionistic)}\\ \textbf{types}}}{\text{ClaType}} \arrow[r, shift left=1.5, "\overset{\textbf{quantization}}{Q}"] \arrow[r, phantom, "\perp"] & \underset{\substack{\textbf{purely}\\ \textbf{quantum}\\ \textbf{(linear)}\\ \textbf{types}}}{\text{QuType}} \arrow[l, shift left=1.5, "\underset{\textbf{classicization}}{C}"] \arrow[loop right, "!\,\textbf{exponential modality}"]
\end{tikzcd} \tag{25}$$

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. 2.9 we obtain (cf. [Ri22a, Prop. 2.1.31]) an exponential modality from the basic type inference provided by a *dependent* linear type theory like `LHoTT` (Lit. 1.8), a possibility first highlighted in [PS12, Ex. 4.2][Sch14a, §4.2].

---

[11] In fact, in `LHoTT` the substructural nature of the linear types is more refined than shown in (24): It *is* possible in `LHoTT` to duplicate the *reference* to terms of linear type, for instance such as to assert their self-identification

$$\mathcal{H} : \mathrm{QuType},\ \psi : \mathcal{H} \quad \vdash \quad \psi = \psi\,,$$

but an accompanying "color palette" ensures that no such duplicate references may be used on the two sides of the tensor product.

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 behavior by *equational* constraints — e.g. for certifying the correctness of quantum teleportation protocols or of quantum error corrections (cf. Rem. 3.2) – one specifically needs certificates of *identification types* (colloquially: "identity types"), certifying the (operational) equality of pairs of data of a given type (cf. Lit. 1.7).

However, 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 (Proto-)`Quipper`, cf. Lit. 1.5. 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. 1.7.

Literature 1.5 (Quantum programming languages). The idea of quantum programming languages formally expressing quantum computational processes (Lit. 1.1) was first systematically expressed in [Kn96], early proposals for formalization are due to [Se04][Val04][SV05][SV09] ("quantum λ-calculus"), [AG05] (`QML`), and [AG10][Gr10] (via "quantum IO", a kind of monadic quantum effects, Lit. 1.17). 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 available NISQ machines, Lit. 1.10), currently there exists essentially one quantum programming language with universal ambition: `Quipper`[12] [GLRSV13][GLRSV13] (for exposition see [Se16]). In its formalized sector called "Proto-`Quipper`" [Ro15, §8][RS18, §4.3] this language may be understood as involving a kind of dependent (Lit. 1.8) linear types, Lit. 1.4) 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 §2.1 we discuss as semantics for the 0-sector (Rem. 1.22) of `LHoTT` (Lit. 1.8).

(Notice that `Quipper` (and `qIO`) are embedded (Lit. 1.6) inside the classical language `Haskell` which means that they lack support for verification of linear (quantum) data types, cf. Lit. 1.4.)

Another quantum programming language scheme with the ambition of certifying (Lit. 1.4) quantum (circuit) programs is `QWIRE`, see [PRZ17][RPZ18][PZ19][RS20][HRHWH21][HRHLH21][ZBSLY23].

Literature 1.6 (Domain-specific embedded programming languages). Besides universal programming languages, more specific tasks – such as quantum circuit programming (cf. Lit. 1.5) – often profit from non-universal languages tailor-made towards the problem at hand – one speaks of *domain-specific languages* (DLS) [Hud98b][Hud98b]. Typically these are *embedded* into ambient universal languages ([Hud96]), by specification of "syntactic sugar" (e.g. [Ra94, §1.6, §1.7, §9]) for blocks of similar code in the ambient language that serve as the building blocks of the domain-specific embedded language.

An example is *do*-notation (Lit. 1.19) for monadic language constructs (Lit. 1.17), and [BHM02, §5.3] suggest that formulating domain-specific embedded languages is close to synonymous with identifying do-notation for suitable monads, citing the example of domain-specific parser languages identified as monadic do-notation by [Wa90, §7.1]. These authors conclude:

> *"Every time a functional programmer designs a combinator library, then, we might as well say that he or she designs a domain specific programming language* [...]. *This is a useful perspective, since it encourages programmers to produce a modular design, with a clean separation between the semantics of the DSL and the program that uses it, rather than mixing combinators and 'raw' semantics willy-nilly. And since monads appear so often in programming language semantics, it is hardly surprising that they appear often in combinator libraries also!*

Existing functional (Lit. 1.16) quantum programming languages such as `qIO` and `Quipper` (Lit. 1.5) are domain-specific languages embedded in `Haskell`, and among these Altenkrich & Green's `qIO` (the *quantumIO-monad*) stands out in its ambition of sticking to the monadic paradigm. However, since the ambient `Haskell` does not verify linear (quantum) data typing (Lit. 1.4, and no other available embedding language did), neither do these embedded languages.

In §3 we aim to show that a nice monadically-embedded quantum programming language with linear tying does exist inside `LHoTT` (Lit. 1.8).

Literature 1.7 (Homotopically typed languages). (For extensive review cf. 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

---

[12] Landing page: `www.mathstat.dal.ca/~selinger/quipper`

other. But taking the idea of program verification by data typing (Lit. 1.4) 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. 1.4) 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. 1.21) 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[13] (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, §5]. Available software that *runs* homotopically typed programs includes `Agda`[14] and `Coq`[15].

Literature 1.8 (Linear homotopically typed languge). Based on the developments of `HoTT` (Lit. 1.7) and in view of the idea of linear data typing for quantum languages (Lit. 1.4) we had previously argued [Sch14a][Sch14b] that there should exist a *linear* enhancement of `HoTT` providing, in addition, a natural formal language for motivic (stable) homotopy (tangent ∞-toposes, Lit. 1.21) and quantum systems. After some partial proposals for such dependent linear type systems ([KPB15][Va15, §3][McB16][Va17][Lu18][Atk18][FKS20][MEO21], see also earlier discussion in [SSt04])[16], a satisfactory *Linear Homotopy Type Theory* (`LHoTT`) has recently been presented by M. Riley [Ri22a], see also [Ri22b][Ri23].

For embedding (Lit. 1.6) the monadic quantum effects of §2 into `LHoTT` all we need is that `LHoTT` verifies the Motivic Yoga (Def. 2.18), which is the case by the discussion in [Ri22a, §2.4].

Literature 1.9 (Topological quantum compilation.). Once serious quantum computation hardware (Lit. 1.3) becomes available, a central effort in quantum computation (Lit. 1.1) concerns *quantum compilation* [MMRP21], namely the translation of high-level quantum algorithms into sequences (circuits) of those logic gates that the hardware actually implements. The seminal *Solovay-Kitaev theorem* ([NC00, 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. 1.3), hence in the case of *topological quantum compilation* (e.g. [HZBS07][Bru14][KBS14]), since here the available gate logic is far remote from then QBit-based operations (17) 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:

(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. 1.7) of which the language `LHoTT` of concern here (Lit. 1.8) is an extension.

Literature 1.10 (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. 1.1) 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 is *topological* protection (Lit. 1.3).

---

[13] 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.

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

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

[16] See [Ri22a, §1.7][Ri22b, p. 22] for critical discussion of these and other previous approaches to dependent linear types.

Literature 1.11 (**Classically controlled quantum computation and dynamic lifting).** The idea of classically controlled quantum computation goes back to [Kn96] and was amplified in [NPW07, §4] (from which we adapted the schematics graphics on p. 4), see also [De14]. The term "dynamic lifting" for the converse control flow (where mid-circuit quantum measurement results are fed back into the classical control logic) is due to [GLRSV13, p. 5], early discussion is in [Ra18, p. 40]; proposals for its categorical semantics are discussed in [RS20][LPVX21][FKRS22a][FKRS22b][CDL22][Lee22].

Of these, the definition in [Lee22, §4.4] of a monad (Lit. 1.17) meant to express dynamic lifting is vaguely in the spirit of the quantum indefiniteness monad $\bigcirc_W$ from §2.3 which in §2.4 we find to express just that: Lee's "lifting monad" applied to a bundle type $\left[\begin{array}{c}\mathcal{H}_\bullet \\ \downarrow \\ W\end{array}\right]$ (in the language of §2.1) produces the bundle type over the set of multisets $[w_i]_{i\in I}$ of elements of W whose fibers are the direct sums $\underset{i\in I}{\oplus}\mathcal{H}_{w_i}$; the idea being to interpret these as the branched Hilbert spaces inside which to locate quantum states obtained after (repeated?) measurement results w_i.

Compare this to the indefiniteness monad, which for a (finite) set of outcomes W sends a pure quantum type $\mathcal{H}$ to $\bigcirc_W\mathcal{H} \equiv \oplus_W\mathcal{H}$ – see the typing of dynamically lifted quantum measurement results on p. 85, and see (220) for the successive lifting of quantum measurements, accumulating the measurements results in the classical context.

1.2 Quantum probability

Literature 1.12 (**Quantum probability and Quantum channels).** Remarkably, in its relation to physical reality, quantum physics (Lit. 1.2) is a *probabilistic* theory ([vN32, §III][MR01]), and yet more remarkably its probabilistic aspect is tied in some deep way to the complex numbers equipped with their involution by complex conjugation:

Hilbert spaces of quantum states. The definition of *Hilbert spaces* $(\mathcal{H}, \langle - | - \rangle)$ in quantum physics ([vN30, §1][vN32, §II.1]) concerns extra structure and properties on the underlying complex vector space of quantum states: (1.) A Hermitian inner product $\langle - | - \rangle$ and (2.) a topological completeness condition. The latter condition is (just) to make sense of infinite-dimensional state spaces and is of no concern for the finite-dimensional Hilbert spaces of interest in quantum information theory (which are automatically complete). The key structure that remains is the Hermitian inner product structure $\langle - | - \rangle$ on a finite-dimensional space $\mathcal{H}$ of quantum states (e.g. [La17, §A.1]), which is (not a complex bilinear on $\mathcal{H} \otimes \mathcal{H}$, but) a *sesquilinear* map, complex-anti linear in the first argument:

$$\underset{\substack{\textbf{Hermtian}\\\textbf{inner product}}}{\langle - | - \rangle} \;:\; \underset{\substack{\textbf{complex conjugate}\\\textbf{space}}}{\overline{\mathcal{H}}} \otimes \underset{\substack{\textbf{complex vector space}\\\textbf{underlying Hilbert space}}}{\mathcal{H}} \longrightarrow \mathbb{C} \tag{26}$$

namely such that

$$\psi, \psi' : \mathcal{H},\;\; c : \mathbb{C} \quad \vdash \quad \begin{array}{ll} \multicolumn{1}{c}{\textbf{Hermitian sesqui-linearity}} & \multicolumn{1}{c}{\textbf{positivity}} \\ \langle \psi' | c \cdot \psi \rangle \;=\; c \, \langle \psi' | \psi \rangle & \langle \psi | \psi \rangle \geq 0\,, \\ \langle \psi | \psi' \rangle \;=\; \overline{\langle \psi' | \psi \rangle}\,, & \langle \psi | \psi \rangle = 0 \;\;\Rightarrow\;\; \psi = 0\,. \\ & \multicolumn{1}{c}{\textbf{non-degeneracy}} \end{array} \tag{27}$$

Bra-Ket notation. The non-degeneracy condition (27) on $\langle - | - \rangle$ means that every element of the linear dual space $\mathcal{H}^* \equiv (\mathcal{H} \multimap \mathbb{C})$ is uniquely of the form $\langle \psi | - \rangle$ for some $\psi \in \mathcal{H}$, which leads to the suggestive *bra-ket* notation traditional in quantum physics (since [Di39], see e.g. [SN94, §1.2][Gri02, §3]):

$$\overset{\textbf{“ket” in Hilbert space}}{|\psi\rangle \;\equiv\; \psi : \mathcal{H}}, \qquad \overset{\textbf{“bra” in dual space}}{\langle \psi | \;\equiv\; \langle \psi | - \rangle \;:\; \mathcal{H}^*}\,. \tag{28}$$

If nothing else, this notation (28) allows one to neatly distinguish between the element $w : W$ in a (finite) set W and the corresponding vector in the linear span $|w\rangle \in \mathrm{Q}W \equiv \underset{W}{\oplus} \mathbb{1}$ (and as such we understand $|-\rangle$ as the return-operation (67) of the “quantization modality” Q, see Def. 2.13 and p. 102). Equipped with the canonical inner product this is an *orthonormal linear basis*:

$$\begin{array}{lccl} \textbf{linear basis} & w : W & \vdash & |w\rangle \;:\; \underset{w:W}{\oplus} \mathbb{C} \;\equiv\; \mathcal{H}\,, \\ \textbf{ortho-normality} & w, w' : W & \vdash & \langle w' | w \rangle \;=\; \delta^{w'}_{w} \;\equiv\; \left\{ \begin{array}{ll} 1 & \text{if } w = w' \\ 0 & \text{otherwise} \end{array} \right. \end{array} \tag{29}$$

More profoundly, the bra-ket notation (28) is a lightweight precursor to the string diagram calculus in dagger-compact closed categories (34) (as amplified by [AC04, §7.2][AC07, p. 6][Co10, §3.3]): For $\mathcal{H}$ a finite-dimensional Hilbert space with orthonormal basis W (29), the vector space of linear maps into some $\mathcal{H}'$ is canonically identified with a space of matrices as follows (136):

$$\begin{array}{ccc} \overset{\substack{\textbf{linear space}\\\textbf{of linear maps}}}{(\mathcal{H} \multimap \mathcal{H}')} & \xrightarrow{\;\;\sim\;\;} & \overset{\substack{\textbf{linear space}\\\textbf{of matrices}}}{\mathcal{H}' \otimes \mathcal{H}^*} \\ \big(\underset{\text{in}}{|w\rangle} \mapsto \underset{\text{out}}{\textstyle\sum_{w'} |w'\rangle A_{w',w}} \big) & \longmapsto & \underset{w,w'}{\sum} \underset{\text{out}}{|w'\rangle} A_{w',w} \underset{\text{in}}{\langle w|} \end{array} \tag{30}$$

The Born rule. The Hermitian inner product $\langle - | - \rangle$ on spaces of quantum states serves to refine the description (21) of the quantum measurement process by assigning a *probability distribution* Prob_ψ to the possible measurement outcomes on a system in state $|\psi\rangle \in \mathcal{H}$ in a state space $\mathcal{H} \simeq \oplus_W \mathbb{C}$ spanned by an orthonormal measurement basis W (29).

The *Born rule* of quantum physics postulates ([Born26, p. 805][Jor27, p. 811][vN32, §III], review in [La09]) that the probability $\mathrm{Prob}_\psi(w)$ for a quantum measurement (21) of a system in a normalized state

$$|\psi\rangle \;:\; \overset{\textbf{normalized states}}{S(\mathcal{H}) \;\equiv\; (|\psi\rangle : \mathcal{H}) \times (\langle \psi | \psi \rangle \;=\; 1)} \tag{31}$$

to yield the result $w : W$ from an orthonormal basis (29) is:

$$\left.\begin{array}{rcl} W & : & \mathrm{FinSet} \\ |\psi\rangle & : & S\big(\underset{w:W}{\oplus}\mathbb{C}\big) \\ w & : & W \end{array}\right\} \vdash \overset{\substack{\textbf{probability to measure } w \\ \textbf{on system in state } |\psi\rangle}}{\mathrm{Prob}_\psi(w)} \quad \overset{\substack{\textbf{equals according} \\ \textbf{to Born's rule}}}{\equiv} \quad \overset{\substack{\textbf{square modulus of} \\ \textbf{transition amplitude}}}{\langle\psi|w\rangle\langle w|\psi\rangle = \overline{\langle w|\psi\rangle}\langle w|\psi\rangle = \Big|\underbrace{\langle w|\psi\rangle}_{\substack{\textbf{“transition amplitude”} \\ \textbf{from } |\psi\rangle \textbf{ to } |w\rangle}}\Big|^2} \tag{32}$$

That the Born rule (32) indeed gives a probability distribution on W is intimately connected to the notion (27) of Hermitian inner products, notably via the corresponding Cauchy-Schwarz inequality:

$$\substack{\textbf{measurement probabilities} \\ \textbf{indeed take values in } [0,1]} \qquad \mathrm{Prob}_\psi(w) \underset{(32)}{\equiv} \big|\langle w|\psi\rangle\big|^2 \underset{\text{Cauchy-Schwarz}}{\leq} \langle w|w\rangle\langle\psi|\psi\rangle \underset{(29)\,(31)}{=} 1$$

$$\substack{\textbf{measurement probs} \\ \textbf{indeed sum to unity}} \quad \sum_w \mathrm{Prob}_\psi(w) \equiv \sum_w \big|\langle w|\psi\rangle\big|^2 = \sum_w \langle\psi|w\rangle\langle w|\psi\rangle = \langle\psi|\underbrace{\Big(\sum_w |w\rangle\langle w|\Big)}_{=\mathrm{id}_{\mathcal{H}}}|\psi\rangle = \langle\psi|\psi\rangle = 1\,.$$

Category theory for Hermitian inner products? The structure of a Hermitian inner product on complex vector spaces (e.g. [KR97, §2.1]), classical as it may be, is somewhat odd (in a precise sense, as we shall see) from the perspective of category theory: On a *real* vector space $\mathcal{V} : \mathrm{Mod}_{\mathbb{R}}$ a (non-degenerate) inner product $\langle\text{-}|\text{-}\rangle$ is a self-duality structure in the category-theoretic sense (cf. [Se12]):

$$\begin{tikzcd} \overset{\substack{\textbf{finite-dimensional} \\ \textbf{vector space}}}{H} \arrow[r, "\sim"] & \overset{\substack{\textbf{its dual} \\ \textbf{vector space}}}{H^*} \\ \psi \arrow[r, mapsto] & \underset{\substack{\textbf{Hermitian} \\ \textbf{inner product}}}{\langle\psi|-\rangle}\,, \end{tikzcd} \tag{33}$$

but for *complex* Hermitian inner product spaces the comparison map (33) is *not complex-linear* — it is complex anti-linear: $c \cdot |\psi\rangle \leftrightarrow \bar{c} \cdot \langle\psi|$. For this reason, finite-dimensional complex Hilbert spaces are *not* the self-dual objects of $\mathrm{Mod}_{\mathbb{C}}$, in contrast to the situation for their real cousins.

Dagger categories. It is ultimately due to this complication (33) that the category-theoretic foundations of quantum information theory have commonly come to be cast in terms of “dagger-categories” (referring, since [Sel07] following [AC04, Prop. 7.3], to the notation “$(\text{-})^\dagger$” for linear operator adjoints; for review see [AC08][Co10][HV12, §2.3, §3.3][Kar18][HV19, §2.3], cf. also [StSt23]), namely by direct axiomatization of the “dagger”-involution on Hom-spaces that is (or would be, in the abstract case) induced by Hermitian inner product structure on the objects:

$$H_1 \xrightarrow{g} H_2 \qquad \vdash \qquad H_1 \xleftarrow{g^\dagger} H_2 \qquad \text{s.t.} \qquad \big\langle g^\dagger(\text{-})\big|-\big\rangle_{H_1} = \big\langle -\big|g(\text{-})\big\rangle_{H_2}\,. \tag{34}$$

In [SS23-QR] we discuss a way of encoding such dagger-structure in `LHoTT`.

Mixed states and density operators. While even a pure quantum state $|\psi\rangle$ (completely characterizing the state of a quantum system, cf. Lit. 1.2) provides only a probabilistic prediction of measurement results given by the Born rule (32), in practice this *objective stochasticity* of nature is accompanied by *subjective stochasticity* due to the fact that the exact quantum state $|\psi\rangle$ of a system may (and typically will) not be known with certainty to the experimenter. Therefore the general state of a quantum system — in the combined sense both of quantum physics and classical statistical physics — is a classical probabilistic *mixture* of quantum states [vN32, §IV.1], or *mixed state* for short (see e.g. [SN94, §3.4][Ish95, §6.1] and particularly [NC00, §2.4][Ku05, §1.4]).

The exact definition notion of what this means was postulated in [vN32, p. 158] and (successfully) used ever since, but is not without conceptual subtlety worthy of consideration: A priori, by a classical mixture of quantum states in a Hilbert space $\mathcal{H}$ one might mean any probability distribution on all of (the underlying set of) the unit sphere $S\mathcal{H}$ of normalized states, or just the projective space $P\mathcal{H}$ of normalized states up to global phase – this would certainly capture some idea of an ensemble of quantum states, but this is *not* what one considers.

Instead, [vN32, p. 157] takes the random measurement collapse (21) as the motivating source of classical uncertainty and thus takes a mixed state to be a probability distribution $p : W \to [0,1]$ on (only) the underlying set W of an orthonormal basis $(|w\rangle : \mathcal{H})_{w:W}$, reflecting the pure states in which one may find the quantum system after W-measurement.

Finally, [vN32, p. 158] observes that it is *technically convenient* (our aim in §2.5 is to motivate this more fundamentally) to encode this probability distribution of basis states as a matrix

$$\overset{\text{probability distribution of basis states}}{\begin{array}{rcl} p_{(-)} \;:\; W & \longrightarrow & [0,1]\,, \sum\limits_{w} p_w = 1 \\ w & \mapsto & p_w \end{array}} \qquad \vdash \qquad \overset{\text{“mixed state” as “density matrix”}}{\rho \;\equiv\; \sum_{w} p_w \cdot |w\rangle\langle w| \;\;:\; \mathcal{H}\otimes\mathcal{H}^*} \tag{35}$$

because then the total probability (of combined quantum and classical origin) to find the system upon quantum measurement of an(other) property W' in the state $|w'\rangle$ is expressed as the *trace* of the operator product of ρ with the projection operator $P_{w'} \equiv |w'\rangle\langle w'|$:

$$\underset{\substack{\text{total prob.}\\ \text{to measure } w'}}{\mathrm{Prob}_\rho(w')} \;=\; \sum_{w} \underset{\substack{\text{classical prob. that}\\ \text{system is in state } |w\rangle}}{p_w} \cdot \underset{\substack{\text{quantum prob. to}\\ \text{measure } w' \text{ in state } |w\rangle}}{\big|\langle w'|w\rangle\big|^2} \;=\; \sum_{w} p_w \langle w'|w\rangle\langle w|w'\rangle \;=\; \langle w'|\Big(\sum_{w} p_w |w\rangle\langle w|\Big)|w'\rangle \;=\; \underset{\substack{\text{trace of density operator}\\ \text{times observable operator}}}{\mathrm{Tr}^{\mathcal{H}}(\rho\cdot P_{w'})} \tag{36}$$

In modern reformulation this means that mixed states are (represented by) *positive* linear operators $\mathcal{H}\to\mathcal{H}$ of unit trace, often called *density operators* or *density matrices* if equivalently understood as elements of $\mathcal{H}\otimes\mathcal{H}^*$ (30):

$$\text{mixed quantum states} \qquad \mathrm{MxdState}(\mathcal{H}) \;\;\equiv\;\; \big(\rho \,:\, \mathcal{H}\otimes\mathcal{H}^*\big)_{\mathrm{undrl}} \times \left(\begin{array}{l} \exists_A\big(\rho \;=\; AA^\dagger\big) \\ \mathrm{Tr}^{\mathcal{H}}(\rho) \;=\; 1 \end{array}\right) \qquad \text{density matrices} \tag{37}$$

This is because the *spectral theorem* for Hermtian operators implies that the positive unit-trace matrices ρ (37) are precisely those which have an eigenbasis W in which their diagonal form is that of (35), with their eigenvalues forming a probability distribution

In particular, the pure states are subsumed among the mixed states as the rank-1 projection operators

$$\overset{\text{pure state}}{|\psi\rangle \;:\; \mathcal{H}} \qquad \vdash \qquad \overset{\text{regarded among mixed states}}{\rho^{|\psi\rangle} \;\equiv\; \frac{|\psi\rangle\langle\psi|}{\big|\langle\psi|\psi\rangle\big|^2}} \;\;:\;\; \mathrm{MxdState}(\mathcal{H})\,. \tag{38}$$

While further examination of this concept shows that it works beautifully and eventually provides a transparent notion of *non-commutative* or *quantum probability* in the algebraic formulation of quantum mechanics (nice review in [Gl09][Gl11]), the curious tensor-doubling involved in passing from the pure state space $\mathcal{H}$ to the density matrices inside $\mathcal{H}\otimes\mathcal{H}^*$ may seem less than obvious from first principles, especially when developing quantum physics from a formal perspective of linear logic (Lit. 1.4). But in §2.5 we observe that $\mathcal{H}\otimes\mathcal{H}^* = \mathcal{H}\otimes(\mathcal{H}\multimap\mathbb{1})$ is naturally understood as the linear version of the costate comonad (118) applied to the tensor unit, and thus in a precise logical sense as the storage of elements of the tensor unit (probability amplitudes) indexable by (pure) quantum states.

Quantum channels. In consequence, where a coherent quantum gate or coherent quantum circuit maps directly

$$\text{pure states}\;\; \mathcal{H}_1 \xrightarrow[\text{unitary map}]{\text{quantum gate}} \mathcal{H}_2 \;\;\text{pure states}$$

between the spaces of pure quantum states (possibly but deterministically parameterized by classical data), a combined quantum and *classically probabilistic* operation on a quantum system — such as incorporating stochastic noise due to a thermal environment — should instead transform the larger space of mixed states (35) or even its ambient linear space of unconstrained matrices:

$$\text{mixed states}\;\; \mathcal{H}_1\otimes\mathcal{H}_1^* \xrightarrow[\substack{\text{completely positive \&}\\ \text{trace-preserving map}}]{\text{quantum channel}} \mathcal{H}_2\otimes\mathcal{H}_2^* \;\;\text{mixed states}\;. \tag{39}$$

but suitably preserving the subspace of density matrices, in that the linear mapping (39):

(i) preserves positivity of operators, in fact it should preserve positivity after coupling to any environment, hence after tensoring with any identity operator (“complete positivity”),

(ii) preserves the trace of operators.

Under these conditions the linear maps (39) are known as *quantum operations* [BZ06, §10][NC00, §8.2] or *quantum channels*[17] [HZ11, §4], expressing the intuition that they reflect the most general physically viable operation on a quantum system, such as when sending its states through a physical communication channel [Wil13][KW20, §3.2].

Since the above two properties may be understood as characterizing the preservation of “quantum probability distributions”; quantum channels may be thought of as the *stochastic maps* in the context of quantum probability theory. If the mapping (39) in addition

---

[17] Since under compact closure (30) the quantum channels (39) are equivalently understood as linear operations on spaces of linear operators $(\mathcal{H}\multimap\mathcal{H})\to(\mathcal{K}\multimap\mathcal{K})$ some authors refer to them as “superoperators” (in the sense of “second order operators”), e.g. [Se04, §6.3]. But besides being ambiguous in itself this term is used with differing conventions by differing authors and might hence better be avoided.

(iii) preserves the identity operator

then one speaks of a *unital quantum channel*, these being the *doubly stochastic maps* in quantum probability.

The fundamental examples of quantum channels are:

- **Unitary quantum channels** (e.g. [HZ11, Ex. 4.6]) corresponding to unitary quantum gates $U : \mathcal{H}_1 \to \mathcal{H}_2$ on pure states and given by conjugation of density matrices with that unitary operator:

  unitary quantum gate as a quantum channel

  $$\begin{array}{rcl} \mathrm{chan}^U : \mathcal{H}_1 \otimes \mathcal{H}_1^* & \longrightarrow & \mathcal{H}_2 \otimes \mathcal{H}_2^* \\ \rho & \longmapsto & U \cdot \rho \cdot U^\dagger \,. \end{array} \tag{40}$$

  This is such that on pure states $\rho^{|\psi\rangle}$ among mixed states (38) the unitary quantum channel acts just as the corresponding quantum gate, in that:

  $$\mathrm{chan}^U \;:\; \rho^{|\psi\rangle} \;\mapsto\; U \cdot \rho^{|\psi\rangle} \cdot U^\dagger \;=\; U \cdot \frac{|\psi\rangle\langle\psi|}{\big|\langle\psi|\psi\rangle\big|^2} \cdot U^\dagger \;=\; \frac{U|\psi\rangle\langle\psi|U^\dagger}{\big|\langle\psi|U^\dagger U|\psi\rangle\big|^2} \;=\; \rho^{U|\psi\rangle} \,.$$

- **Mixed unitary quantum channels** are probabilistic ensembles of unitary channels (40) in that they are given by S-tuples $(U_s : \mathcal{H}_1 \to \mathcal{H}_2)_{i:S}$ of unitary operators indexed over an inhabited finite index-set S, and by a probability distribution $p_{(-)} : S \to [0,1]$, as

  classical mixture of unitary quantum gates as a quantum channel

  $$\begin{array}{rcl} \mathrm{chan}^{(U_\bullet, p)} : \mathcal{H}_1 \otimes \mathcal{H}_1^* & \longrightarrow & \mathcal{H}_2 \otimes \mathcal{H}_2^* \\ \rho & \longmapsto & \sum\limits_{s:S} p_s \, U_s \cdot \rho \cdot U_s^\dagger \,. \end{array} \tag{41}$$

  For example, the **bit-flip quantum channel** is the mixed unitary channel (41) on single qbit states $\mathrm{QBit} \equiv \underset{\{0,1\}}{\oplus} \mathbb{C}$ (166) given for $p \in [0,1]$ by (e.g. [NC00, §8.1 & 8.3.3]):

  qbit-flip quantum channel

  $$\begin{array}{rcl} \mathrm{flip}_p : \mathrm{QBit} \otimes \mathrm{QBit}^* & \longrightarrow & \mathrm{QBit} \otimes \mathrm{QBit}^* \\ \rho & \longmapsto & (1-p)\rho \,+\, p\, X \cdot \rho \cdot X \,, \end{array} \tag{42}$$

  where $X \equiv |0\rangle\langle 1| + |1\rangle\langle 0|$ is the "Pauli X" quantum gate (or *quantum NOT gate*) which swaps (flips) the two canonical qbit-basis elements.

  Hence the bit-flip quantum channel (42) models a process where a qbit the flipped with probability p and retained as is with probability $(1-p)$. This is a simple model for the effect of *quantum noise*.

- **Measurement quantum channels** with respect to an orthonormal linear basis $\mathcal{H} \simeq \oplus_W \mathbb{C}$ (29), given by

  measurement statistics as a quantum channel

  $$\begin{array}{rcl} \mathrm{chan}^W : \mathcal{H} \otimes \mathcal{H}^* & \longrightarrow & \mathcal{H} \otimes \mathcal{H}^* \\ \rho & \longmapsto & \sum_w P_w \, \rho \, P_w \end{array} \tag{43}$$

  (where $P_w \equiv |w\rangle\langle w|$). This description (43) of quantum measurement is originally due to [Lü51, (8)] and has become standard quantum physics lore (a nice discussion is in [Wh12]): Notice that the density matrix on the right of (43) expresses a *classical uncertainty regarding which measurement result was obtained* and instead provides the probabilistic mixture of collapsed quantum states for all possible measurement outcomes, weighted according to the Born rule (32):

  quantum measurement channel on a pure quantum state...

  $$\begin{array}{rcl} |\psi\rangle : S\big(\underset{w:W}{\oplus}\mathbb{C}\big) \quad \vdash \quad \mathrm{chan}^W : \rho^{|\psi\rangle} & \mapsto & \sum\limits_w P_w \cdot \rho^{|\psi\rangle} \cdot P_w \\ & = & \sum\limits_w |w\rangle\langle w|\psi\rangle\langle\psi|w\rangle\langle w| \\ & = & \sum\limits_w |w\rangle \, \mathrm{Prob}_\psi(w) \, \langle w| \,. \end{array}$$

  ...produces the mixture of all possible measurement outcomes weighted by their Born probability

  Incidentally, (43) is not the only sensible modeling of quantum measurement (21) on mixed states: If we do know and record which specific $w : W$ has been measured, then the typing should rather be:

  measurement of mixed states with dynamic lifting of results

  $$\begin{array}{rcl} \mathcal{H} \otimes \mathcal{H}^* & \longrightarrow & (W \to \mathbb{C}) \\ \rho & \longmapsto & (w \mapsto P_w \cdot \rho \cdot P_w) \end{array} \tag{44}$$

  This was in fact Lüders' first proposal: [Lü51, (7)]! In a quantum protocol, this description (44) of the measurement process retains the probabilities of the measurement outcomes but "dynamically lifts" (1.11) the actual outcome to a new classical parameter (Lit. 1.11). We naturally recover this description (44) as a monoidal-monad operation, below in (221).

Later it was noticed [JZ85] that (43) may be understood as arising from the **decoherence** of the quantum state upon its coupling to an environment (here: the measurement apparatus), by which *the off-diagonal elements of the density matrix vanish* in the measurement basis ([JZ85, (3.57)], cf. [Om94, p.277][Schl07, p. 95][Schl19, (7)]):

$$
\begin{aligned}
\overset{\text{pure state}}{\rho^{|\psi\rangle} \;\equiv\; |\psi\rangle\langle\psi|} \;=\; \Big(\sum_{w}|w\rangle\langle w|\psi\rangle\Big)\Big(\sum_{w'}\langle\psi|w'\rangle\langle w'|\Big) \quad & = \quad \sum_{w,w'} |w\rangle\overbrace{\langle w|\psi\rangle\langle\psi|w'\rangle}^{\text{coherent phases}}\langle w'| \\
& \underset{\substack{\text{measurement}\\ \text{channel}}}{\xmapsto{\;\mathrm{chan}^W\;}} \quad \underbrace{\sum_{w} \; |w\rangle\langle w|\psi\rangle\langle\psi|w\rangle\langle w|}_{\text{decohered mixed state}} \;=\; \sum_{w} p_w \cdot |w\rangle\langle w| \,.
\end{aligned}
\tag{45}
$$

- **Averaging quantum channels** operate on a compound quantum systems $\mathcal{H}\otimes\mathcal{B}$ — the system $\mathcal{H}$ of primary interest coupled to an *environment* or *thermal bath* $\mathcal{B}$ — by retaining of the environment only the *expectation value* of its effects on the system, which means to form the partial trace of density matrices over $\mathcal{B}$:

$$
\begin{array}{c}
\text{averaging over a subsystem}\\ \text{as a quantum channel}
\end{array}
\quad
\begin{array}{rcl}
\mathrm{chan}^{\mathcal{B}} \;:\; (\mathcal{H}\otimes\mathcal{B})\otimes(\mathcal{H}\otimes\mathcal{B})^* \;\tilde{\to}\; \mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* & \longrightarrow & \mathcal{H}\otimes\mathcal{H}^* \\
|\psi,\beta\rangle\langle\beta',\psi'| & \longmapsto & |\psi\rangle\langle\beta'|\beta\rangle\langle\psi'| \,.
\end{array}
\tag{46}
$$

An elementary but profound insight into the structure of quantum physics — often referred to under the term **decoherence** – is the observation that quantum measurement channels (43) may be understood as nothing but the composite of a unitary evolution (40) of the system $\mathcal{H}$ *coupled* to its environment $\mathscr{B}$ by way of a deterministic measuring process, but then followed by an averaging (46) over the exact state of the measurement device:

Concretely, if $|b_{\mathrm{ini}}\rangle \,:\, \mathcal{B}$ denotes the initial state of a "device" then any notion of this device measuring the system $\mathcal{H}$ (in its measurement basis W) under their joint unitary quantum evolution should be reflected in a unitary operator under which the system $\mathcal{H}$ remains invariant *if* it is purely in any eigenstate $|w\rangle$ of the measurement basis, while in this case the measuring system evolves to a corresponding "pointer state" $|b_w\rangle$ [Zu81, (1.1)][JZ85, (1.1)] (following [vN32, §VI.3], review includes [Schl07, (2.52)]):

$$
\begin{array}{rcl}
U_W \;:\; \mathcal{H}\otimes\mathcal{B} & \xrightarrow{\substack{\text{unitary}\\ \text{measurement process}}} & \mathcal{H}\otimes\mathcal{B} \\
|w, b_{\mathrm{ini}}\rangle & \longmapsto & |w, b_w\rangle
\end{array}
\tag{47}
$$

for b_{ini} and b_w distinct elements of an (in practice: approximately-)orthonormal basis for $\mathcal{B}$. (There is always a unitary operator with this mapping property (47), for instance the one which moreover maps $|w, b_w\rangle \mapsto |w, b_{\mathrm{ini}}\rangle$ and is the identity on all remaining basis elements.) But then the composition of the corresponding unitary quantum channel with the averaging channel over $\mathcal{B}$ is indeed equal to the W-measurement channel (cf. e.g. [Schl07, (2.117)], going back to [Zeh70, (7)]):

$$
\begin{tikzcd}[column sep=large]
\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\substack{\text{couple system to}\\ \text{initialized meas. device}\\ \mathrm{chan}^{b_{\mathrm{ini}}}}"] \arrow[rrr, bend left=20, "\substack{\text{quantum measurement channel}\\ \mathrm{chan}^W}"] & \mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[r, "\substack{\text{evolve system \& device}\\ \text{under meas. interaction}\\ \mathrm{chan}^{U_W}}"] & \mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[r, "\substack{\text{average over states}\\ \text{of measuring device}\\ \mathrm{chan}^{\mathcal{B}}}"] & \mathcal{H}\otimes\mathcal{H}^* \\
\underbrace{\sum_{w,w'}\rho_{w,w'}|w\rangle\langle w'|}_{\rho} \arrow[r, phantom, "\longmapsto"] & \sum_{w,w'}\rho_{w,w'}|w, b_{\mathrm{ini}}\rangle\langle b_{\mathrm{ini}}, w'| \arrow[r, phantom, "\longmapsto"] & \sum_{w,w'}\rho_{w,w'}|w, b_w\rangle\langle b_{w'}, w'| \arrow[r, phantom, "\longmapsto"] & \sum_{w,w'}\rho_{w,w'}|w\rangle\langle b_{w'}|b_w\rangle\langle w'| \\
 & & = & \sum_{w,w'}\rho_{w,w'}|w\rangle\delta_w^{w'}\langle w'| \\
 & & = & \underbrace{\sum_{w}\rho_{w,w}|w\rangle\langle w|}_{\sum_w P_w\cdot\rho\cdot P_w}
\end{tikzcd}
\tag{48}
$$

- **Coupling channels** (rarely made explicit as such, but conceptually important to notice) which for any mixed state ρ_{env} of a given system $\mathcal{B}$ form the tensor product state:

$$
\begin{array}{c}
\text{coupling to ancillary system}\\ \text{as a quantum channel}
\end{array}
\quad
\begin{array}{rcl}
\mathrm{chan}^{\rho_{\mathrm{env}}} \;:\; \mathcal{H}\otimes\mathcal{H}^* & \longrightarrow & (\mathcal{H}\otimes\mathcal{B})\otimes(\mathcal{B}\otimes\mathcal{H})^* \\
\rho & \longmapsto & \rho\otimes\rho_{\mathrm{env}}
\end{array}
\tag{49}
$$

Operator-sum decomposition of quantum channels. The fundamental theorem of quantum channel theory characterizes them ([Ch75], review in [NC00, Thm. 8.1][Wil13, Thm. 4.4.1]) as exactly those linear maps of the form

$$\begin{array}{ccc} \mathcal{H}_1 \otimes \mathcal{H}_1^* & \longrightarrow & \mathcal{H}_2 \otimes \mathcal{H}_2^* \\ \rho & \longmapsto & \sum\limits_r E_r \cdot \rho \cdot E_r^\dagger \end{array} \tag{50}$$

for non-empty tuples of linear operators

$$R \,:\, \mathrm{FinSet}, \quad r \,:\, R \quad \vdash \quad E_r \,:\, \mathcal{H}_1 \to \mathcal{H}_2 \qquad \text{s.t.} \left\{ \begin{array}{lcll} \sum_r \, E_r^\dagger \cdot E_r & = & \mathrm{id} & \text{(preservation of trace)} \\ \sum_r \, E_r \cdot E_r^\dagger & = & \mathrm{id} & \text{(for unital channels)} \; . \end{array} \right.$$

This looks like a purely technical lemma, but it has profound conceptual consequences, such as the following:

Environmental representation of quantum channels. Remarkably, quantum endo-channels $\mathrm{chan} : \mathscr{H} \otimes \mathscr{H}^* \to \mathscr{H} \otimes \mathscr{H}^*$ may alternatively be characterized as those linear maps which arise – in generalization of the situation for measurement channels (48) – under the 3-step procedure of:

(i) *coupling* the system ρ to an environment system $\mathscr{B}$ in some state ρ_{env} (49),
(ii) *evolving* the compound system $\rho \otimes \rho_{\mathrm{env}}$ through a unitary quantum channel chan^U (40)
(iii) *averaging* the result over the environmental states (46):

$$\begin{tikzcd}[column sep=large] \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\overset{\substack{\textbf{couple system to}\\ \textbf{initialized meas. device}}}{\mathrm{chan}^{\rho_{\mathrm{env}}}}"] \arrow[rrr, bend left, "\overset{\textbf{any quantum channel}}{\mathrm{chan}}"] & (\mathcal{H} \otimes \mathcal{B}) \otimes (\mathcal{B} \otimes \mathcal{H})^* \arrow[r, "\overset{\substack{\textbf{evolve system \& environment}\\ \textbf{under some interaction}}}{\mathrm{chan}^{U_W}}"] & (\mathcal{H} \otimes \mathcal{B}) \otimes (\mathcal{B} \otimes \mathcal{H})^* \arrow[r, "\overset{\substack{\textbf{average over states}\\ \textbf{of environment}}}{\mathrm{chan}^{\mathcal{B}}}"] & \mathcal{H} \otimes \mathcal{H}^* \\ \rho \arrow[r, mapsto] & \rho \otimes \rho_{\mathrm{env}} \arrow[r, mapsto] & U_{\mathrm{tot}}(\rho \otimes \rho_{\mathrm{env}})U_{\mathrm{tot}}^\dagger \arrow[r, mapsto] & \mathrm{Tr}^{\mathscr{B}}(U_{\mathrm{tot}}(\rho \otimes \rho_{\mathrm{env}})U_{\mathrm{tot}}^\dagger) \end{tikzcd} \tag{51}$$

That all such averaged environment-interactions are quantum channels is immediate from the three component steps being quantum channels. That every quantum channel has an environmental representation (originally remarked by [Li75, inside Lem. 5]) follows by choosing an operator-sum decomposition (50): Then taking $\mathscr{B} \equiv \oplus_r \mathbb{C}$, singling out one of its basis vectors $|r_{\mathrm{ini}}\rangle$ as the pure environmental state

$$\rho_{\mathrm{env}} \;\equiv\; |r_{\mathrm{init}}\rangle\langle r_{\mathrm{ini}}| \,, \tag{52}$$

and finally observing that any unitary operator of the form

$$\begin{array}{rccc} U \;:\; & \mathcal{H} \otimes \mathcal{B} & \longrightarrow & \mathcal{H} \otimes \mathcal{B} \\ & |\psi\rangle \otimes |r_{\mathrm{ini}}\rangle & \longmapsto & \sum\limits_r E_r |\psi\rangle \otimes |r\rangle \end{array}$$

serves the purpose (e.g. [NC00, p. 365][Att, Thm. 6.7][BZ06, §10.4]).

(The ontological import of this theorem is profound: It is consistent to assume that the world at large fundamentally evolves according to deterministic unitary evolution of pure quantum states, while all apparent classical stochasticity in the evolution of small subsystems results entirely from ignorance about the exact microstate of their quantum environment.)

Noisy/unistochastic/DQC quantum channels. While every quantum channel is environmentally realized (51) as a bath-average of a unitary evolution of the given system coupled to a *pure* state of the environment (52), some quantum channels are realized even by coupling to mixed environmental states.

In the extreme but (practically highly) relevant case where the coupling is to an environment in its maximally mixed (namely uniformly distributed) quantum state (55) some authors speak of *noisy quantum operations* [HHO03][MHP19] others of *unistochastic quantum channels* [ZB04, p. 259][BZ06][MKZ13]:

$$\begin{tikzcd}[column sep=large] \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\overset{\substack{\textbf{couple system to}\\ \textbf{maximally mixed bath}}}{\mathrm{chan}^{\mathrm{unif}}}"] \arrow[rrr, bend left, "\overset{\textbf{unistochastic quantum channel}}{\mathrm{chan}}"] & (\mathcal{H} \otimes \mathcal{B}) \otimes (\mathcal{B} \otimes \mathcal{H})^* \arrow[r, "\overset{\substack{\textbf{evolve system \& environment}\\ \textbf{under some interaction}}}{\mathrm{chan}^{U_W}}"] & (\mathcal{H} \otimes \mathcal{B}) \otimes (\mathcal{B} \otimes \mathcal{H})^* \arrow[r, "\overset{\substack{\textbf{average over states}\\ \textbf{of environment}}}{\mathrm{chan}^{\mathcal{B}}}"] & \mathcal{H} \otimes \mathcal{H}^* \\ \rho_{\mathrm{sys}} \arrow[r, mapsto] & \rho_{\mathrm{sys}} \otimes \rho_{\mathscr{B}}^{\mathrm{unif}} \arrow[r, mapsto] & U_{\mathrm{tot}}(\rho_{\mathrm{sys}} \otimes \rho_{\mathscr{B}}^{\mathrm{unif}})U_{\mathrm{tot}}^\dagger \arrow[r, mapsto] & \mathrm{Tr}^{\mathscr{B}}\big(U_{\mathrm{tot}}(\rho_{\mathrm{sys}} \otimes \rho_{\mathscr{B}}^{\mathrm{unif}})U_{\mathrm{tot}}^\dagger\big) \end{tikzcd} \tag{53}$$

But the same idea underlies already the model of quantum computation introduced under the abbreviation *DQC1* by [KL98][PLMP03][SJ08] (also known as the "one clean qbit"-model), motivated by the (noisy) reality of quantum computation (specifically on NMR spin-resonance qbits). In this case $\mathcal{H} \equiv \mathrm{QBit}$ is a single QBit, and one initializes the system in state $|0\rangle$ (say) and measures the expectation value (61) of the observable $O_{P_0} \equiv |0\rangle\langle 0|$ (60) in the output of the above channel (53), given by the following formula (cf. [SJ08, (1)]):

probability measured by (repeated) DQC1 computations

$$p_0 \;=\; \mathrm{Tr}^{\mathrm{QBit}}\Big(P_0 \cdot \mathrm{Tr}^{\mathcal{B}}\big(U_{\mathrm{tot}}(|0\rangle\langle 0| \otimes \rho^{\mathrm{unif}}_{\mathcal{B}})U^\dagger_{\mathrm{tot}}\big)\Big). \tag{54}$$

The relation of this DQC1 model to unistochastic quantum channels is obvious but has been made explicit only recently [XCGX23, §III] (and not using the "unistochastic" terminology). We give a natural monadic typing in Ex. 16.

Incidentally, we may observe that among all coupling channels (49), those which couple to the *maximally mixed state* of the environment this way, namely the one represented by a multiple of the identity matrix and representing the *uniform* probability distribution on (any set of) orthonormal basis states $(|b\rangle)_{b:B}$ are *dual* (in a precise sense) to the averaging channels (46):

uniformly distributed mixture of bath states

$$\rho^{\mathrm{unif}}_{\mathcal{B}} \;\equiv\; \frac{1}{\dim(\mathcal{B})}\,\mathrm{id}_{\mathcal{B}} \;=\; \sum_b \frac{|b\rangle\langle b|}{\dim(\mathcal{B})} \;:\; \mathcal{B}\otimes\mathcal{B}^* \tag{55}$$

coupling to uniform bath as a quantum channel

$$\begin{array}{rccc}
\mathrm{chan}^{\rho^{\mathrm{unif}}_{\mathcal{H}}} \;:\; & \mathcal{H}\otimes\mathcal{H}^* & \xrightarrow{\qquad\qquad\quad} & (\mathcal{H}\otimes\mathcal{B})\otimes(\mathcal{B}\otimes\mathcal{H})^* \\
& \rho_{\mathrm{sys}} & \longmapsto & \rho_{\mathrm{sys}}\otimes\rho^{\mathrm{unif}}_{\mathcal{B}}
\end{array} \tag{56}$$

In §2.5 we understand this dual pair of quantum channels as the initial (terminal) cases among the (co)monadic QuantumState (co)monad transformations.

For example, every *uniformly* mixed unitary quantum channel (41) (i.e., one in which every unitary operator U_s appears with the same probability $1/\mathrm{Card}(S)$) is unistochastic (53), with coupled-unitary given as shown below:

uniformly mixed unitary quantum gates as a quantum channel

$$\begin{array}{rccl}
\mathrm{chan}^{(U_\bullet)} \;:\; & \mathcal{H}_1\otimes\mathcal{H}_1^* & \xrightarrow{\qquad\quad} & \mathcal{H}_2\otimes\mathcal{H}_2^* \\
& \rho & \longmapsto & \sum\limits_{s:S}\tfrac{1}{\mathrm{Card}(S)}\,U_s\cdot\rho\cdot U_s^\dagger\,,
\end{array} \tag{57}$$

$$\begin{array}{rccc}
U_{\mathrm{tot}} \;:\; & \mathcal{H}\otimes\bigoplus\limits_S\mathbb{C} & \xrightarrow{\qquad\qquad\quad} & \mathcal{H}\otimes\bigoplus\limits_S\mathbb{C} \\
& |\psi\rangle\otimes|s\rangle & \longmapsto & U_s|\psi\rangle\otimes|s\rangle\,.
\end{array} \tag{58}$$

In fact, on single qbits, every mixed unitary actually has such a uniformly mixed unitary presentation [MHP19, Thm. 1.2] and hence is unistochastic (53).

For example, with the general argument given in [MHP19, Lem. 1.1] one finds that a **unistochastic presentation of the bit-flip channel** (42) is given by the following total unitary (58) on the single qbit-system coupled to an environment consisting of one other qbit:

unistochastic environmental realization of bit-flip quantum channel

$$\begin{array}{rccr}
U^{\mathrm{flip}_p}_{\mathrm{tot}} \;:\; & \mathrm{QBit}\otimes\mathrm{QBit} & \xrightarrow{\qquad\qquad\qquad} & \mathrm{QBit}\otimes\mathrm{QBit} \\
& |0\rangle\otimes|b\rangle & \longmapsto & \cos(\phi/2)\,|0\rangle\otimes|b\rangle \\
& & & +(-1)^b\,\mathrm{i}\sin(\phi/2)\,|1\rangle\otimes|b\rangle \\
& |1\rangle\otimes|b\rangle & \longmapsto & (-1)^b\,\mathrm{i}\sin(\phi/2)\,|0\rangle\otimes|b\rangle \\
& & & +\cos(\phi/2)\,|1\rangle\otimes|b\rangle\,,
\end{array}
\quad \begin{array}{l}\text{where} \\ \phi = \arccos(1-2p).\end{array} \tag{59}$$

Closely related to quantum channels:
Quantum observables are much like quantum channels to the trivial system, but without the requirement that the trace be preserved:

$$\begin{array}{ccc}
\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow[O_A]{\text{quantum observable}} & \mathbb{1} \\
|\psi\rangle\langle\phi| & \longmapsto & \langle\phi|A|\psi\rangle
\end{array}
\quad\leftrightarrow\quad
\begin{array}{ccc}
\mathcal{H} & \xrightarrow[\text{positive operator}]{\qquad\qquad\qquad} & \mathcal{H} \\
|\psi\rangle & \longmapsto & \underbrace{a^\dagger a}_{A}\,|\psi\rangle
\end{array} \tag{60}$$

In particular, given a mixed state represented by a density matrix $\rho : \mathcal{H}\otimes\mathcal{H}^*$ from (35), then the *expectation value* of an observable O_A (60) in this state is the value of the quantum operation O_A on ρ, which equals the trace (36) of the operator product of the associated operator A with the density matrix:

$$\overset{\substack{\text{expectation value of}\\ \text{observabled } O_A\\ \text{in mixed state } \rho}}{\langle O_A\rangle_\rho \;\equiv\; O_A(\rho)} \;=\; \overset{\substack{\text{trace of product of}\\ \text{associated operator } A\\ \text{with density matrix } \rho}}{\mathrm{Tr}(A\cdot\rho)}\,. \tag{61}$$

This means that after passing through a unitary quantum channel chan^U (40) an observable O_A is transformed according to the *Heisenberg evolution formula* (e.g. [BGL95, p. 36][Pre04, (3.44)])

$$O_A \;\longmapsto\; \mathrm{chan}^U(O_A) \;\equiv\; O_{U\cdot A\cdot U^\dagger} \tag{62}$$

in that

$$\langle \mathrm{chan}^U(O_A)\rangle_{\mathrm{chan}^U(\rho)} \;\equiv\; \mathrm{Tr}((U\cdot A\cdot U^\dagger)\cdot(U\cdot\rho\cdot U^\dagger)) \;=\; \mathrm{Tr}(A\cdot\rho) \;=\; \langle A\rangle_\rho\,.$$

1.3 Monadic effects

Literature 1.13 (**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*." With 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]):

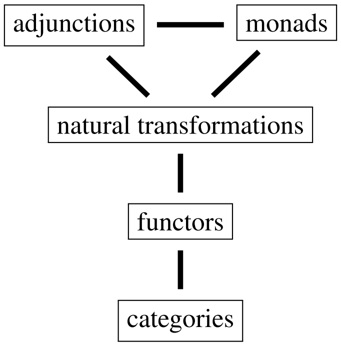
$$
\overset{\substack{\textbf{A proposition } P_\bullet \\ \textbf{about/dependent on} \\ \textbf{the possible worlds } w}}{\begin{array}{rccc} P_\bullet : & W & \longrightarrow & \mathrm{Prop} \\ & w & \longmapsto & P_w \end{array}}
\quad \overset{\textbf{yields}}{\vdash} \quad
\overset{\substack{\textbf{The proposition } \Box_W P \\ \textbf{that } P_\bullet \textbf{ holds } \textit{necessarily}\textbf{, namely} \\ \textbf{in/for } \textit{all} \textbf{ worlds } w' \textbf{ that appear} \\ \textbf{as possible as the given one } w}}{\begin{array}{rccc} \Box_W P : & W & \longrightarrow & \mathrm{Prop}, \\ & w & \longmapsto & \underset{\substack{(w':W)\times \\ R(w,w')}}{\forall} P_{w'} \end{array}}
\quad \textbf{and} \quad
\overset{\substack{\textbf{The proposition } \Diamond_W P \\ \textbf{that } P \textbf{ holds } \textit{possibly}\textbf{, namely} \\ \textbf{in/for } \textit{some} \textbf{ world } w' \textbf{ that appears} \\ \textbf{as possible as the given one } w}}{\begin{array}{rccc} \Diamond_W P : & W & \longrightarrow & \mathrm{Prop} \\ & w & \longmapsto & \underset{\substack{(w':W)\times \\ R(w,w')}}{\exists} P_{w'} \end{array}}
\tag{63}
$$

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][Kob97] and the resulting relation of modalities to (computational effect-)monads in computer science (Lit. 1.17) 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 §2) that, therefore, any dependent type theory may equivalently be regarded as (epistemic) *modal type theory* (Lit. 1.14) seems not to have received attention until the note [*n*Lab14] (cf. [Cor20, Ch. 4]). We expand on this novel point of view in the main text around Thm. 2.23.

Literature 1.14 (**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. 1.13) to *modal type theory*. After leading a niche existence for some time, the amplification [Sch13, §3.1][ScSh14] of *cohesive* modalities (see [ss20-Orb]) 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 (are definable by *admissible rules* inferred from just the inference rules of the dependent linear types). This fact is what drives our observation that `LHoTT` (Lit. 1.8) already knows about quantum measurement effects – the feature just has to be brought out by meticulous syntactic sugaring (Lit. 1.6).

Literature 1.15 (**Category theory).** The point of category theory ([ML71/97][Ke82][Bor94b]) has been said to be the notion of *natural transformations* between mathematical stuctures, where the concept of *categories* themselves just serves to allow for speaking about *functors* which in turn are the subjects of these natural transformations. This is implicit in the title and introduction of Eilenberg & MacLane's original [EM45], and made more explicit Freyd in [Fr64, p. 1]. But this is really only half of the story.

Namely natural transformations, in turn, support the concept of *adjunctions* between categories, and *this* is where category theory becomes a theory: Adjunctions and their many equivalent incarnations such as (Kan extensions, (co)limits, (co)terminality and notably) monads (for which see Lit. 1.17) are the fundamental mathematical phenomena where category theory provides its non-trivial theorems. (Curiously, adjunctions are arguably the formalization of *dualities*, hence it is not misleading to say that category theory is really the *theory of duality*. In fact, [EM45] motivate their introduction of category theory with the example of dualizable objects, see (133)).



Literature 1.16 (**Functional programming languages).** In programming, it is a familiar idea (expanded on in Lit. 1.4) that every *datum* d be of some specified *data type* D, denoted "$d : D$". This being so, then 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 data types*:

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

(64)

In the simplest examples (cf. p. 12), the semantics of the simplest functional

- *classical languages* may be in the *category of sets*,
  where elementary programs are interpreted as *logic gates*

$$\mathrm{Bit}^{\times^{n_{\mathrm{in}}}} \longrightarrow \mathrm{Bit}^{\times^{n_{\mathrm{out}}}}$$

- *quantum languages* may be in the category of $\mathbb{C}$-*vector spaces*,
  where elementary programs are interpreted as *quantum gates*

$$\mathrm{QBit}^{\otimes^{n_{\mathrm{in}}}} \longrightarrow \mathrm{QBit}^{\otimes^{n_{\mathrm{out}}}}$$

The point of *functional programming* (e.g. [Th96][Th91]) is that programs are such functions and *nothing but* such functions of data (compiled under function composition), in that they have:

- no side-effects – besides producing their declared output,
- no context-dependence — besides on their declared input,

on the global state of the computing environment.

Therefore one also speaks of *pure functions* or *pure programs*, for emphasis. This is in contrast to more traditionally 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 declaration of their input/output data types. In contrast, the purity of functional programs is what makes them completely deterministic, hence predictable by mathematical analysis and hence formally verifiable (Lit. 1.4).

This relation between (i) computation (ii) data typing and (iii) categorical algebra turns out to be so tight as to effectively exhibit three equivalent perspectives on the same underlying structure, a remarkable phenomenon that has been called the *computational trilogy* (for pointers see [SS22, p. 4]):

$$\begin{tikzcd}
\overset{①}{\boxed{\textbf{Computation}}} \arrow[rr, leftrightarrow, bend left] \arrow[dr, leftrightarrow, bend right] & & \overset{②}{\boxed{\textbf{Type Theory}}} \arrow[dl, leftrightarrow, bend left] \\
& \overset{③}{\boxed{\textbf{Alg. Topology}}} &
\end{tikzcd} \tag{65}$$

Of course, in practice one needs programs which *do* cause side-effects, or which *do* have context-dependence. Noticing the above qualifications, these may absolutely be described by functional programs, but

side-effects are to be *declared* as part of the output data type,
context-dependence is to be *declared* as part of the input data type.

In line with the computational trilogy (65), there should be fundamental concepts in type theory and in categorical algebra which correspond to such effect/context-declaration in typed programming languages. Indeed, these correspondent ares the very concepts of *modalities* (Lit. 1.14) and of *monads*, see Lit. 1.17.

$$\begin{tikzcd}
\overset{①}{\boxed{\textbf{Effects}}} \arrow[rr, leftrightarrow, bend left] \arrow[dr, leftrightarrow, bend right] & & \overset{②}{\boxed{\textbf{Modalities}}} \arrow[dl, leftrightarrow, bend left] \\
& \overset{③}{\boxed{\textbf{Monads}}} &
\end{tikzcd} \tag{66}$$

Literature 1.17 (Computational Effects and Monadic 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. 1.14).

Computational effects... The idea ([Mog89a][Mog89][Wa90][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][Wi22, §10]) via `bind`- and `return`-operations, as follows:

$$\begin{tikzcd}
\textbf{first program} & & \textbf{second program} & & \textbf{pure data} \\
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathcal{E}(D_2) & D_2 \arrow[r, "\mathrm{prog}_{23}"] & \mathcal{E}(D_3) \arrow[dd, bend left=60, "\textbf{bind previous effects into second program}"] & d : D \arrow[dd, bend left=60, "\textbf{return pure data with trivial effect}"] \\
\textbf{with output data of nominal type } D_2 \textbf{ but causing effects of type } \mathcal{E}(-) & & \textbf{with input data of type } D_2 & & \\
& & \mathcal{E}(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(D_3) \arrow[ddd, bend left=60, "\textbf{compose}"] & \mathtt{return}^{\mathcal{E}}_{D}(d) : \mathcal{E}(D) \\
& & \textbf{carry any previous } \mathcal{E}(-)\textbf{-effects along} & & \mathcal{E}\textbf{-effectful data} \\
& \mathcal{E}\textbf{-composite program} & & & \\
D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathcal{E}(D_2) \arrow[rr, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & & \mathcal{E}(D_3) & \\
& \textbf{causing cumulative } \mathcal{E}(-)\textbf{-effects} & & &
\end{tikzcd} \tag{67}$$

such that

$$\begin{tikzcd}
 & \text{binding effect to trivial effect} & & & \text{trivial effect...} & & \text{...bound into program} & \\
\mathcal{E}(D) \arrow[rr, "\mathtt{bind}^{\mathcal{E}}\ \mathtt{return}^{\mathcal{E}}_{D}"] \arrow[rr, bend right, "\mathrm{id}_{\mathcal{E}}"'] & & \mathcal{E}(D) & & D_1 \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D}"] \arrow[rr, bend right, "\mathrm{prog}_{12}"'] & \mathcal{E}(D_1) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{12}"] & \mathcal{E}(D_2) \\
 & \text{is to carry it along identically} & & & & \text{...has no effect on program} &
\end{tikzcd}$$

$$\begin{tikzcd}
\text{bound effects} & \mathtt{bind}^{\mathcal{E}}\big(D_1 \arrow[r, "\mathrm{prog}_{12}"] \arrow[d, equal] & \mathcal{E}(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(D_3)\big) \arrow[d, equal] \\
\text{get carried along} & \mathcal{E}(D_1) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{12}"] & \mathcal{E}(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(D_3)
\end{tikzcd} \tag{68}$$

One also speaks of *Kleisli composition* (in honor of [Kl65, p. 545]) and writes ("fish notation", e.g. [Mi19, p. 321]):

$$\mathrm{prog}_{12} >\!\!=\!\!> \mathrm{prog}_{23} \quad \equiv \quad (\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}) \circ \mathrm{prog}_{12} \tag{69}$$

...as monads on the type system. 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")

$$\begin{tikzcd}
\mathcal{E} \;:\; \mathrm{Type} \arrow[rr, "\text{functor underlying monad}"] & & \mathrm{Type} \\
(D_1 \xrightarrow{f} D_2) \arrow[rr, mapsto, "\text{regard } f \text{ as effectless program}"'] & & \mathtt{bind}^{\mathcal{E}}\big(D_1 \xrightarrow{f} D_2 \xrightarrow{\mathtt{return}^{\mathcal{E}}_{D}} \mathcal{E}(D_2)\big)
\end{tikzcd} \tag{70}$$

which carries the structure of a **monad**[18] (cf. [ML71/97, §VI][Bor94b, §4], older terminology: "triple"), namely natural transformations

$$D : \mathrm{Type} \quad \vdash \quad D \xrightarrow[]{\overset{\text{monad unit/return}}{\mathrm{ret}^{\mathcal{E}}_{D} \,\equiv\, \mathtt{return}^{\mathcal{E}}_{D}}} \mathcal{E}(D)\,, \qquad \mathcal{E}(\mathcal{E}(D)) \xrightarrow[]{\overset{\text{monad product/join}}{\mathrm{join}^{\mathcal{E}}_{D} \,\equiv\, \mathtt{bind}^{\mathcal{E}}\mathrm{id}_{\mathcal{E}(D)}}} \mathcal{E}(D) \tag{71}$$

satisfying the axioms of a unital monoid object (139), in that they make the following natural diagrams commute

$$\begin{tikzcd}
\mathcal{E}(D) \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[d, "\mathcal{E}(\mathrm{ret}^{\mathcal{E}}_{D})"'] \arrow[dr, equal, "\text{unitality}" description] & \mathcal{E}(\mathcal{E}(D)) \arrow[d, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"] & & \mathcal{E}(\mathcal{E}(\mathcal{E}(D))) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[d, "\mathcal{E}(\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)})"'] \arrow[dr, phantom, "\text{associativity}"] & \mathcal{E}(\mathcal{E}(D)) \arrow[d, "\mathrm{join}^{\mathcal{E}}_{D}"] \\
\mathcal{E}(\mathcal{E}(D)) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{\mathcal{E}(D)}"'] & \mathcal{E}(D)\,, & & \mathcal{E}(\mathcal{E}(D)) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{D}"'] & \mathcal{E}(D)\,.
\end{tikzcd} \tag{72}$$

Namely conversely, given such a monad the bind-operation on some prog : $D_1 \to \mathcal{E}(D_2)$ is recovered as:

$$\begin{tikzcd}
\text{already effectful data} & \text{program produces further effects} & \text{doubly effectful data} & \text{effects joined together} & \text{plain effectful data} \\
\mathcal{E}(D_1) \arrow[rr, "\mathcal{E}(\mathrm{prog})"] \arrow[rrrr, bend right, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}"', "\text{bind previous effects into program}"' {yshift=-12pt}] & & \mathcal{E}(\mathcal{E}(D_2)) \arrow[rr, "\mathrm{join}^{\mathcal{E}}_{D_2}"] & & \mathcal{E}(D_2)\,,
\end{tikzcd} \tag{73}$$

which shows that the `join`-operation is that which *joins consecutive effects into a single effect*, whence then terminology.

Monads induced by adjunctions. Monads arise from (cf. [ML71/97, §VI.1][Bor94b] – and also give rise to, see (98) below) – *adjoint functors* ("adjunctions" between categories, cf. [ML71/97, §IV]), namely pairs of back-and-forth functors (here: between categories of types)

---

[18] The terminology "monad" for (70) 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" of (the Kleisli category of) the monad by an effect handler (89).

$$\begin{tikzcd}
\mathrm{Type}' \arrow[rr, phantom, "\perp"] \arrow[rr, shift right=2, "\underset{\textbf{right adjoint}}{R}"'] && \mathrm{Type} \arrow[ll, shift right=2, "\overset{\textbf{left adjoint}}{L}"'] \arrow[loop right, "R\circ L \,\equiv:\, \mathcal{E} \quad \textbf{induced monad}"]
\end{tikzcd} \tag{74}$$

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

$$\mathrm{Hom}_{\mathrm{Type}}(-,\, R(-)) \xleftrightarrow{\;\widetilde{(-)}\;} \mathrm{Hom}_{\mathrm{Type}'}(L(-),\, -) \tag{75}$$

and (equivalently) with natural transformations

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

$$\begin{array}{ccc}
\overset{\textbf{adjunction unit}}{\Big(D \xrightarrow{\mathrm{ret}^{RL}_D} R\circ L(D)\Big)} & \longmapsfrom & \overset{\textbf{identity}}{\Big(L(D) \xrightarrow{\mathrm{id}_{L(D)}} L(D)\Big)} \\[1em]
\underset{\textbf{identity}}{\Big(R(D') \xrightarrow{\mathrm{id}_{R(D')}} R(D')\Big)} & \longmapsto & \underset{\textbf{adjunction counit}}{\Big(L\circ R(D') \xrightarrow{\mathrm{obt}^{LR}_{D'}} D'\Big)}
\end{array}$$

satisfying the *zig-zag identities*

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

from which the monad structure (71) on $\mathcal{E} := R\circ L$ is obtained as follows:

monad unit/return is... **monad product/join is...**

$$\begin{tikzcd}
D \arrow[r, "\mathrm{ret}^{\mathcal{E}}_D"] \arrow[d, phantom, "\rotatebox{90}{$\equiv$}"] & \mathcal{E}(D) \arrow[d, phantom, "\rotatebox{90}{$\equiv$}"] \\
D \arrow[r, "\mathrm{ret}^{RL}_D"'] & R\circ L(D)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{E}(\mathcal{E}(D)) \arrow[r, "\mathrm{join}^{\mathcal{E}}_D"] \arrow[d, phantom, "\rotatebox{90}{$\equiv$}"] & \mathcal{E}(D) \arrow[d, phantom, "\rotatebox{90}{$\equiv$}"] \\
R\circ \underbrace{L\circ R}\circ L(D) \arrow[r, "R\big(\mathrm{obt}^{LR}_{L(D)}\big)"'] & R\circ L\,.
\end{tikzcd} \tag{76}$$

...the adjunction unit / return **...value under R of adjunction counit/obtain on value under L**

Typing of effects via Strong monads. As a technical aside, beware that in describing effect 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 §3 below), the return- and bind-operations (67) will be typed *not* externally as

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

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

$$\begin{aligned}
\mathtt{return}^{\mathcal{E}}_D \;:\; D \to \mathcal{E}(D)\,, \qquad \mathtt{bind}^{\mathcal{E}}_{D_1,\,D_2} \;&:\; (D_1 \to \mathcal{E}(D_2)) \to (\mathcal{E}(D_1) \to \mathcal{E}(D_2)) \\
&=\; \mathcal{E}(D_1) \times (D_1 \to \mathcal{E}(D_2)) \to \mathcal{E}(D_2) \\
&=\; \mathcal{E}(D_1) \to \big((D_1 \to \mathcal{E}(D_2)) \to \mathcal{E}(D_2)\big)\,,
\end{aligned} \tag{77}$$

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 (e.g. [LS86, §I][Bor94b, §6.1]). (Here we stick to notation for cartesian monoidal structure just for the purpose of exposition, see (162) 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 [Ra12, §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 (24) it needs to be established (which we do in Prop. 2.7). 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. 2.3 — with braiding transformations

$$\mathrm{braid}^{\otimes}_{D,\,D'} \;:\; D \otimes D' \longrightarrow D' \otimes D$$

then *symmetric monoidal* structure on a monad $\mathcal{E}$ ([Ko70, p. 8], cf. e.g. [Se13, §1.2])[19]

structure

$$\begin{tikzcd}
\mathcal{E}(D)\otimes\mathcal{E}(D') \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'}"] \\
\mathcal{E}(D\otimes D')
\end{tikzcd}$$

monad

$$\begin{tikzcd}
D\otimes D' \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{D}\otimes\mathrm{ret}^{\mathcal{E}}_{D'}"] \arrow[d, equal] & \mathcal{E}(D)\otimes\mathcal{E}(D') \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'}"] & (\mathcal{E}\circ\mathcal{E}(D))\otimes(\mathcal{E}\circ\mathcal{E}(D')) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{D}\otimes\mathrm{join}^{\mathcal{E}}_{D'}"] \arrow[d, "\mathcal{E}(\mathrm{pair}^{\mathcal{E}}_{D,D'})\circ\mathrm{pair}^{\mathcal{E}}_{\mathcal{E}(D),\,\mathcal{E}(D')}"] & (\mathcal{E}(D))\otimes(\mathcal{E}(D')) \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'}"] \\
D\otimes D' \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{D\otimes D'}"] & \mathcal{E}(D\otimes D') & \mathcal{E}\circ\mathcal{E}(D\otimes D') \arrow[r, "\mathrm{join}^{\mathcal{E}}_{D\otimes D'}"] & \mathcal{E}(D\otimes D')
\end{tikzcd}$$

monoidal

$$\begin{tikzcd}
& \mathcal{E}(\mathbb{1})\otimes\mathcal{E}(D) \arrow[dr, "\mathrm{pair}^{\mathcal{E}}_{\mathbb{1},D}"] & & & \mathcal{E}(\mathbb{1})\otimes\mathcal{E}(D) \arrow[dr, "\mathrm{pair}^{\mathcal{E}}_{D,\mathbb{1}}"] & & \mathcal{E}(D)\otimes\mathcal{E}(D')\otimes\mathcal{E}(D'') \arrow[r, "\mathrm{id}\otimes\mathrm{pair}^{\mathcal{E}}_{D',D''}"] \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'}"] & \mathcal{E}(D)\otimes\mathcal{E}(D'\otimes D'') \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'\otimes D''}"] \\
\mathbb{1}\otimes\mathcal{E}(D) \arrow[ur, "\mathrm{ret}^{\mathcal{E}}_{\mathbb{1}}\otimes\mathrm{id}"] & = & \mathcal{E}(D) & \mathcal{E}(D)\otimes\mathbb{1} \arrow[ur, "\mathrm{id}\otimes\mathrm{ret}^{\mathcal{E}}_{\mathbb{1}}"] & = & \mathcal{E}(D) & \mathcal{E}(D\otimes D')\otimes\mathcal{E}(D) \arrow[r, "\mathrm{pair}^{\mathcal{E}}_{D\otimes D',D''}"] & \mathcal{E}(D\otimes D'\otimes D'')
\end{tikzcd} \tag{78}$$

symmetric

$$\begin{tikzcd}
\mathcal{E}(D)\otimes\mathcal{E}(D') \arrow[r, "\mathrm{braid}^{\otimes}_{\mathcal{E}(D),\mathcal{E}(D')}"] \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D,D'}"] & \mathcal{E}(D')\otimes\mathcal{E}(D) \arrow[d, "\mathrm{pair}^{\mathcal{E}}_{D',D}"] \\
\mathcal{E}(D\otimes D') \arrow[r, "\mathcal{E}(\mathrm{braid}^{\otimes}_{D,\,D'})"'] & \mathcal{E}(D'\otimes D)
\end{tikzcd}$$

bijectively induces "commutative" strong monad structure ([Ko72, Thm. 2.3], detailed review in [GLLN08, §7.3, §A.4] [Ra12, Prop. 3.3.9]) hence in particular the required enriched monad structure (77).

Examples of effect monads. Fundamental examples of effect monads in classical computer science (and in their linear version of profound importance to us in §2) include (cf. [Mog91, Ex. 1.1]):

- The **reader-** or **environment-monad** (e.g. [Mi19, §21.2.3][Uu21, p. 22], we use "W" for the *worlds* being read out, cf. Lit. 1.13):

$$\begin{array}{rcccc} W\mathrm{Read} & : & \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ & & D & \longmapsto & [W,\,D] \end{array} \tag{79}$$

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

$$\begin{tikzcd}
\textbf{comonoid } W \\ \text{(ambient data)} & W\times W & & W \arrow[ll, "\mathrm{diag}_W"'] \arrow[r, "\exists!"] & * \\
W\textbf{-reader monad} & {[W,\,[W,\,D]]} \simeq {[W\times W,\,D]} \arrow[rr, "\mathrm{join}^{W\mathrm{Read}}_{D}", "\equiv [\mathrm{diag}_W,\,D]"'] & & {[W,\,D]} & {[*,\,D]} \simeq D \arrow[l, "\mathrm{ret}^{W\mathrm{Read}}_{D}"', "\equiv \mathrm{const}"]
\end{tikzcd} \tag{80}$$

  Hence a W-Reader-effectful program is one whose nominal output is *indefinite* (195) until a global parameter $w : W$ is read in, and the handling of W-Reader-effects is the handing-along of this global parameter.

$$\begin{aligned}
\mathtt{bind}^{W\mathrm{Read}}_{D,\,D'} \;&: (D \to (W \to D')) \xrightarrow{\begin{subarray}{c}\textbf{binding of}\\ W\textbf{Reader effects}\end{subarray}} ((W \to D) \to (W \to D')) \\
\mathtt{bind}^{W\mathrm{Read}}_{D,\,D'} \;&\equiv \underset{\begin{subarray}{c}\textbf{program producing output}\\ \textbf{depending on}\\ \textbf{a global } W\textbf{-parameter}\end{subarray}}{\Big(d \mapsto (w \mapsto d'_w(d))\Big)} \mapsto \Big((w \mapsto d_w) \mapsto (\,w \;\mapsto\; d'_w(d_w))\Big)
\end{aligned}$$

global parameter **gets passed to** **all subsequent** **programs**

---

[19] We assume without restriction [Schau01] that the monoidal structure $\otimes$, $\mathbb{1}$ is "strict", i.e. that its unitors and associators are identity morphisms, whence we do not show then in these diagrams.

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

$$\begin{tikzcd} A\mathrm{Write} : \mathrm{Type} \arrow[r] & \mathrm{Type} \\ D \arrow[r, mapsto] & A \times D\,. \end{tikzcd} \tag{81}$$

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

$$\begin{tikzcd} \substack{\textbf{monoid } W \\ \text{(data output stream)}} & A \times A \arrow[r, "\mathrm{prod}_A"] & A & * \arrow[l, "\mathrm{unit}_A"'] \\ A\textbf{-writer monad} & A \times A \times D \arrow[r, "\mathrm{join}_D^{A\mathrm{Writ}} \equiv", "\mathrm{prod}_A \times \mathrm{id}_D"'] & A \times D & * \times D = D \arrow[l, "\mathrm{ret}_D^{A\mathrm{Write}} \equiv"', "\mathrm{unit}_A \times \mathrm{id}_D"] \end{tikzcd} \tag{82}$$

(Here the unitality and associativity properties of the monoid structure on A are evidently equivalent to the corresponding properties (72) of the associated writer monad.) In typical applications A is a *free monoid* on an alphabet, hence is the type of *strings* of such characters with join product 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)

$$\begin{aligned} \mathtt{bind}^{A\mathrm{Write}}_{D,D'} \;&:\; (D \to A \times D') \longrightarrow (A \times D \to A \times D') \\ \mathtt{bind}^{A\mathrm{Write}}_{D,D'} \;&\equiv\; (d \mapsto (a_d,\, d'_d)) \mapsto ((a,\, d) \mapsto (\underset{\textbf{concatenated logs}}{a \cdot a_d},\, d'_d)) \end{aligned}$$

- The **state monad** (e.g. [BHM02, Ex. 42][PP02, §3][Mi19, §21.2.5][Uu21, 1, p. 24])

$$\begin{tikzcd} W\mathrm{State} : \mathrm{Type} \arrow[r] & \mathrm{Type} \\ D \arrow[r, mapsto] & {[W,\, W \times D]} \end{tikzcd} \tag{83}$$

given by

$$\begin{tikzcd} {[W,\; W \times [W,\; W \times D]]} \arrow[r, "\mathrm{join}_D^{W\mathrm{State}}"] & {[W,\; W \times D]} & D \arrow[l, "\mathrm{ret}_D^{W\mathrm{State}}"'] \\ f \arrow[r, mapsto] & \mathrm{ev}(f(\text{-})) & \\ & (w \mapsto (w, d)) & d \arrow[l, mapsto] \end{tikzcd} \tag{84}$$

hence with bind-operation as follows:

$$\begin{aligned} \mathtt{bind}^{W\mathrm{State}}_{D_1,D_2} \;&:\; \big(D_1 \to (W \to W \times D_2)\big) \to \big((W \to W \times D_1) \to (W \to W \times D_2)\big) \\ \mathtt{bind}^{W\mathrm{State}}_{D_1,D_2} \;&\equiv\; \qquad \mathrm{prog} \;\mapsto\; \big((w \mapsto (w'_w, d_w)) \;\mapsto\; (w \mapsto \mathrm{prog}(d_w)(w'_w))\big)\,. \end{aligned} \tag{85}$$

Such WState-effectful programs are adjoint (75) to programs of the form (20)

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

(also known as *Mealy machines* following [Me55], see e.g. [Pa03, §1.1.3] for the modern formulation and [OM16, p. 262][PK23, p. 3] for our state-effectful perspective) which may be understood as producing their nominal output only after *reading 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[20] for a *random access memory* ("RAM", see [Ya19, p. 26 & Fig. 1.10]):

$$\begin{tikzcd} D \arrow[r, "W\textbf{-RAM effectful program}"] & {[W,\, W \times D']} & \substack{\textbf{type of} \\ W\textbf{State-effectful } D'\textbf{-data}} \\ \underset{\substack{\textbf{nominal} \\ \textbf{input data}}}{d} \arrow[r, mapsto] & \Big(\underset{\textbf{RAM readout}}{w} \mapsto \big(\underset{\textbf{RAM rewrite}}{w'_{(w,d)}},\, \underset{\substack{\textbf{nominal} \\ \textbf{output data}}}{d'_{(w,d)}}\big)\Big) & \end{tikzcd} \tag{86}$$

---

[20] 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" (see Rem. 2.27 below), which are the modales over the dual of the state monad: The co-state co-monad [O'C11].

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 Msg : Type some type of error messages, the exception monad is

$$\mathrm{Exc}_{\mathrm{Msg}} \quad : \quad \begin{array}{ccc} \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ D & \longmapsto & D \sqcup \mathrm{Msg} \end{array} \tag{87}$$

whose return-operation is the coprojection into coproduct and whose join operation 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, which 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 ([PP13]):

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

consistency conditions

(88)

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

$$\begin{tikzcd}[column sep=huge]
\mathcal{E}(D) \arrow[r, "\substack{\text{monad action on modale}\\ \rho \,\equiv\, \mathtt{handle}^{\mathcal{E}}_{D}\mathrm{id}_D}"] & D
\end{tikzcd} \tag{89}$$

satisfying the axioms of a monoid module (143), in that it makes the following squares commute:

$$\begin{tikzcd}
D \arrow[d, "\eta_D"'] \arrow[dr, bend left, "\mathrm{id}"] \arrow[dr, phantom, "\mathrm{utl}_{\mathcal{E}}(\rho)"' near start] & \\
\mathcal{E}(D) \arrow[r, "\rho"'] & D
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\mathcal{E}(\mathcal{E}(D)) \arrow[r, "\mathcal{E}(\rho)"] \arrow[d, "\mu_D"] \arrow[dr, phantom, "\mathrm{act}_{\mathcal{E}}(\rho)"] & \mathcal{E}(D) \arrow[d, "\rho"] \\
\mathcal{E}(D) \arrow[r, "\rho"'] & D\,.
\end{tikzcd} \tag{90}$$

unitality — action property

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

$$\begin{tikzcd}
\mathcal{E}(D_1) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\rho_1"] & \mathcal{E}(D_2) \arrow[d, "\rho_2"] \\
D_1 \arrow[r, "f"] & D_2\,,
\end{tikzcd} \tag{91}$$

which we will indicate by the following notation (which is non-standard but nicely suggestive):

$$\begin{tikzcd}[column sep=huge]
\substack{\mathcal{E}\text{-modale}\\ \text{stucture}}\; D_1 \arrow[loop above, "\mathcal{E}"] \arrow[r, "f", "\text{modale homomorphism}"'] & D_2 \;\substack{\mathcal{E}\text{-modale}\\ \text{stucture}} \arrow[loop above, "\mathcal{E}"]
\end{tikzcd} \tag{92}$$

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, $\mathrm{Type}_{\mathcal{E}}$:

$$\begin{tikzcd}
 & \textbf{free construction} & \substack{\textbf{free } \mathcal{E}\textbf{-modales in } \mathrm{Type} \\ \text{(``Kleisli category'')}} & \textbf{total comparison functor} & \substack{\mathcal{E}\textbf{-modales in } \mathrm{Type} \\ \text{(``Eilenberg-Moore category'')}} \\
\mathrm{Type} \arrow[rr, two heads, "F_{\mathcal{E}}"] \arrow[rrrr, bend right, "F^{\mathcal{E}}"'] & & \mathrm{Type}_{\mathcal{E}} \arrow[rr, hook, "K_{U^{\mathcal{E}}F^{\mathcal{E}}}"] & & \mathrm{Type}^{\mathcal{E}} \\
\{B : \mathrm{Type}\} & & \{\mathcal{E}(B),\, \rho_B \equiv \mu_B : \mathcal{E}(\mathcal{E}(B)) \to \mathcal{E}(B)\} & & \{D : \mathrm{Type},\, \rho : \mathcal{E}(D) \to D \,\big|\, \mathrm{untl}_{\mathcal{E}}(\rho), \mathrm{act}_{\mathcal{E}}(\rho)\}
\end{tikzcd} \tag{93}$$

Incidentally, notice that thereby every $\mathcal{E}$-effect handler ρ (90) is itself a modale-homomorphism (91) from a free modale (93):

$$\begin{tikzcd}
\substack{\textbf{free modale} \\ \textbf{structure}} & \mathcal{E}(D) \arrow[loop above, "\mathcal{E}"] \arrow[rr, "\substack{\textbf{given effect handler} \\ \rho \equiv \mathtt{handle}^{\mathcal{E}}_{D}}", "\textbf{modale homomorphism}"'] & & D \arrow[loop above, "\mathcal{E}"] & \substack{\textbf{given modale} \\ \textbf{structure}}
\end{tikzcd} \tag{94}$$

(and regarding it this way is crucial for the monadic typing of quantum measurement, see p. 81 below).

Concretely, the *Kleisli equivalence* re-identifies the homomorphism between free $\mathcal{E}$-modales with the $\mathcal{E}$-effectful programs that we started with (67), as follows (e.g. [Bor94b, Prop. 1.4.6]):

$$\begin{tikzcd}
\mathrm{Type}_{\mathcal{E}} \arrow[rr, hook] & & \mathrm{Type}^{\mathcal{E}} \\
D \arrow[rr, phantom, "\longmapsto"] & & (\mathcal{E}(D),\, \mu_D) \\
\mathrm{Type}_{\mathcal{E}}(D, D') \arrow[rr, leftrightarrow, "\sim", "\textbf{Kleisli equivalence}"'] & & \mathrm{Type}^{\mathcal{E}}\Big((\mathcal{E}(D), \mu_D),\, (\mathcal{E}(D'), \mu_{D'})\Big) \\
\Big(D \xrightarrow{\quad f \quad} \mathcal{E}(D')\Big) \arrow[rr, phantom, "\longmapsto"] & & \Big(\mathcal{E}(D) \xrightarrow{\mathcal{E}(f)} \mathcal{E}(\mathcal{E}(D')) \xrightarrow{\mu_{D'}} \mathcal{E}(D')\Big) \\
\Big(D \xrightarrow{\mathrm{ret}^{\mathcal{E}}_{D}} \mathcal{E}(D) \xrightarrow{\phi} \mathcal{E}(D')\Big) \arrow[rr, phantom, "\longleftarrow\!\!\shortmid"] & & \Big(\mathcal{E}(D) \xrightarrow{\quad \phi \quad} \mathcal{E}(D')\Big).
\end{tikzcd} \tag{95}$$

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

$$\begin{tikzcd}
\mathrm{Type}_{\mathcal{E}} \arrow[r, hook, "K_{U^{\mathcal{E}}F^{\mathcal{E}}}"] \arrow[rr, bend left, "U_{\mathcal{E}}"] & \mathrm{Type}^{\mathcal{E}} \arrow[r, "U^{\mathcal{E}}", "\textbf{forgetful functor}"'] & \mathrm{Type} \\
 & (D, \rho : \mathcal{E}(D) \to D) \arrow[r, mapsto] & D
\end{tikzcd} \tag{96}$$

and both adjunctions $F_{\mathcal{E}} \dashv U_{\mathcal{E}}$ and $F^{\mathcal{E}} \dashv U^{\mathcal{E}}$ re-induce (74) the original monad, with the modale structure recovered from the adjunction counit obt (e.g. [ML71/97, §VI.2, Thm. 1, §IV.5, Thm. 1]):

$$\begin{tikzcd}
 & & U^{\mathcal{E}}F^{\mathcal{E}}U^{\mathcal{E}}(D,\rho) \arrow[r, equal] \arrow[d, "U^{\mathcal{E}}(\mathrm{obt}_{(D,\rho)})"'] & \mathcal{E}(D) \arrow[d, "\rho"] \\
(D,\rho) \;:\; \mathrm{Type}^{\mathcal{E}} & \vdash & & \\
 & & U^{\mathcal{E}}(D,\rho) \arrow[r, equal] & D\,.
\end{tikzcd} \tag{97}$$

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. [ML71/97, §VI.3]):

$$\begin{tikzcd}[column sep=huge]
 & & & \mathrm{Type}_{\mathcal{E}} \arrow[dd, hook, "K_{UF}", "\substack{\textbf{initial} \\ \textbf{comparison} \\ \textbf{functor}}"'] \arrow[ddll, shift left, "U_{\mathcal{E}}"] & \substack{\textbf{free } \mathcal{E}\textbf{-modales in } \mathrm{Type} \\ \text{(``Kleisli category'')}} \\
 & & & & \\
\textbf{induced monad} & \mathcal{E} \circlearrowright \mathrm{Type} \arrow[uurr, two heads, shift left, "B \mapsto (\mathcal{E}(B), \rho \equiv \mu_B)", "F_{\mathcal{E}}"'] \arrow[rr, shift left, "F"] \arrow[ddrr, shift left, "F^{\mathcal{E}}"] & & \mathrm{Type}' \arrow[ll, shift left, "U"] \arrow[dd, "K^{UF}", "\substack{\textbf{terminal} \\ \textbf{comparison} \\ \textbf{functor}}"'] & \textbf{any adjunction} \text{ for } \mathcal{E} \\
 & & & & \\
 & & & \mathrm{Type}^{\mathcal{E}} \arrow[uull, shift left, "U^{\mathcal{E}}", "\textbf{monadic adjunction}"'] & \substack{\mathcal{E}\textbf{-modales in } \mathrm{Type} \\ \text{(``EM-category'')}}
\end{tikzcd} \tag{98}$$

The monadicity theorem (cf. [Bor94b, Thm. 4.4.4]) characterizes the monadic adjunctions on the bottom of diagram (98): For a functor U to be *monadic* in that it is of the form $U^{\mathcal{E}}$ in (98), it is sufficient[21] that

---

[21] The necessity clause involves the preservation of those coequalizers that are "split", 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:

$$\Rightarrow \qquad \begin{tikzcd} \mathcal{D} \arrow[d, shift left=1.5ex, "U"] \arrow[d, phantom, "\dashv"] \\ \mathrm{Type} \arrow[u, shift left=1.5ex, "F"] \end{tikzcd} \ \text{is monadic} \qquad \Rightarrow \qquad \begin{tikzcd} \mathcal{D} \arrow[rr, "\sim"] \arrow[dr, shift left, "U"] \arrow[dr, phantom, "\dashv"] & & \mathrm{Type}^{\mathcal{E}} \arrow[dl, shift left, "U^{\mathcal{E}}"] \\ & \mathrm{Type} \arrow[ul, shift left, "F"] \arrow[ur, shift left, "F^{\mathcal{E}}"] \arrow[ur, phantom, "\dashv"] \arrow[loop below, "\mathcal{E}"] & \end{tikzcd} \tag{99}$$

Relative monads. While monads are equivalent to computational effects as in (67), the latter notion has a suggestive generalization to what are called *relative monads* [ACU15](see also [AMcD23]), where the effect-attaching functor $\mathcal{E}$ (70) is not an endofunctor but maps between two different categories of types

$$\mathcal{E} \;:\; \mathrm{Type} \to \mathrm{Type}'\,.$$

An common situation is where $\mathrm{Type} \hookrightarrow \mathrm{Type}'$ is a subcategory inclusion, where it just means that $\mathcal{E}$-effects are attachable only to types in this subcategory. Generally one can consider any comparison functor

$$J \;:\; \mathrm{Type} \to \mathrm{Type}' \tag{100}$$

and define a *J-relative monad* structure to be given by *J-relative* return- and bind-operations:

$$\begin{array}{lcl} D \;:\; \mathrm{Type} & \vdash & \mathtt{return}^{\mathcal{E}}_{D} \;:\; J(D) \to \mathcal{E}(D) \\[10pt] D_1, D_2 \;:\; \mathrm{Type} & \vdash & \mathtt{bind}^{\mathcal{E}}_{D_1, D_2} \;:\; \big(J(D_1) \to \mathcal{E}(D_2)\big) \longrightarrow \big(\mathcal{E}(D_1) \to \mathcal{E}(D_2)\big) \end{array} \tag{101}$$

otherwise satisfying the same form of consistency conditions as in the non-relative case (68).

As a simple but relevant example, for every actual monad $\mathcal{E}'$ on Type', its precomposition with any functor $J \;:\; \mathrm{Type} \to \mathrm{Type}'$ (100) yields a J-relative monad ([ACU15, Prop. 2.3]) via:

$$\mathcal{E} \;\equiv\; \mathcal{E} \circ J\,, \qquad \mathtt{return}^{\mathcal{E}}_{D} \;\equiv\; \mathtt{return}^{\mathcal{E}'}_{J(D)}\,, \qquad \mathtt{bind}^{\mathcal{E}}_{D_1, D_2} \;\equiv\; \mathtt{bind}^{\mathcal{E}'}_{J(D_1), D_2}\,. \tag{102}$$

Monad transformations. With monads encoding effectful programs, one is bound to consider several monadic effects $\mathcal{E}$, $\mathcal{E}'$, ... at once, and procedures that *transform* these into each other:

$$D \;:\; \mathrm{Type} \qquad \vdash \qquad \mathtt{trans}^{\mathcal{E} \to \mathcal{E}'}_{D} \;:\; \mathcal{E}(D) \longrightarrow \mathcal{E}'(D)\,. \tag{103}$$

For consistency these transformations (103) ought to respect the return- and bind-operations (67), in that the following diagrams commute:

$$D \;:\; \mathrm{Type} \qquad \vdash \qquad \begin{tikzcd} D \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D}"] \arrow[rr, bend right, "\mathtt{return}^{\mathcal{E}'}_{D}"] & \mathcal{E}(D) \arrow[r, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D}"] & \mathcal{E}'(D) \end{tikzcd}$$

$$\begin{array}{l} \mathrm{prog}_{12} \;:\; D_1 \to \mathcal{E}(D_2) \\ \mathrm{prog}_{23} \;:\; D_2 \to \mathcal{E}(D_3) \end{array} \quad \vdash \quad \begin{tikzcd} D_1 \arrow[r, "\mathrm{prog}_{12}"] \arrow[d, equal] & \mathcal{E}(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(D_3) \arrow[r, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_3}"] & \mathcal{E}'(D_3) \arrow[d, equal] \\ D_1 \arrow[r, "\mathrm{prog}_{12}"] & \mathcal{E}(D_2) \arrow[r, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}"] & \mathcal{E}'(D_2) \arrow[r, "\mathtt{bind}^{\mathcal{E}'}\left(D_2 \xrightarrow{\mathrm{prog}_{23}} \mathcal{E}(D_3) \xrightarrow{\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_3}} \mathcal{E}'(D_3)\right)"] & \mathcal{E}'(D_3)\,, \end{tikzcd} \tag{104}$$

hence such that the Kleisli composition (69) is respected:

$$(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ \mathrm{prog}_{12}) \;\texttt{>=>}\; (\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_3}\circ \mathrm{prog}_{23}) \quad=\quad \mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ(\mathrm{prog}_{12}\;\texttt{>=>}\;\mathrm{prog}_{23})\,, \tag{105}$$

exhibiting a covariant functor on free modales (93)

$$\begin{tikzcd}
 & \mathrm{Type} \arrow[dl, "F_{\mathcal{E}}"'] \arrow[dr, "F_{\mathcal{E}'}"] & \\
\mathrm{Type}_{\mathcal{E}} \arrow[rr] & & \mathrm{Type}_{\mathcal{E}'}
\end{tikzcd}$$

$$\mathcal{E}(D_1)\xrightarrow{\phi}\mathcal{E}(D_2) \qquad\longmapsto\qquad \mathtt{bind}^{\mathcal{E}'}\Big(D_1 \xrightarrow{\mathtt{return}^{\mathcal{E}}_{D_1}} \mathcal{E}(D_1)\xrightarrow{\phi}\mathcal{E}(D_2)\xrightarrow{\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}}\mathcal{E}'(D_2)\Big) \tag{106}$$

which preserves (as indicated on top) the free maps (93), $\phi = \mathcal{E}(f) \mapsto \mathcal{E}'(f)$, due to the commutativity of the following pasting diagram (the left square being the unitality condition in (104), the right square the implied naturality property (107)):

$$\begin{tikzcd}
D_1 \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D_1}"] \arrow[d, equal] & \mathcal{E}(D_1) \arrow[r, "\mathcal{E}(f)"] \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_1}"] & \mathcal{E}(D_2) \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}"] \\
D_1 \arrow[r, "\mathtt{return}^{\mathcal{E}}_{D_1}"] & \mathcal{E}'(D_1) \arrow[r, "\mathcal{E}'(f)"] & \mathcal{E}'(D_2)\,.
\end{tikzcd}$$

This notion of *monad transformers* originates with [Esp95, §2.6], the explicit definition (104) is due to [LHJ95, p. 339] now commonly used in `Haskell`[22]. But we may observe that the equivalent definition not in terms of the bind- but the join-operation (considered within `Haskell` in [SPWJ19, §2.2]) is much older:

Namely in category theory, a *morphism of monads* is known to be a natural transformation of their underlying functors (70)

$$\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'} \;:\; \mathcal{E}\longrightarrow\mathcal{E}' \tag{107}$$

which is compatible with the return- and join-operations (71) as follows:

$$D\,:\,\mathrm{Type} \quad\vdash\quad \begin{tikzcd}
D \arrow[r, equal] \arrow[d, "\mathrm{ret}^{\mathcal{E}}_D"'] & D \arrow[d, "\mathrm{ret}^{\mathcal{E}'}_D"] \\
\mathcal{E}(D) \arrow[r, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_D"] & \mathcal{E}'(D)\,,
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{E}\circ\mathcal{E}(D) \arrow[r, "\mathcal{E}\left(\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_D\right)"] \arrow[d, "\mathrm{join}^{\mathcal{E}}_D"'] & \mathcal{E}\circ\mathcal{E}'(D) \arrow[r, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_{\mathcal{E}'(D)}"] & \mathcal{E}'\circ\mathcal{E}'(D) \arrow[d, "\mathrm{join}^{\mathcal{E}'}_D"] \\
\mathcal{E}(D) \arrow[rr, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_D"] & & \mathcal{E}'(D)\,.
\end{tikzcd} \tag{108}$$

Notice here that the order of the composites at the top right of (108) is arbitrary, since the naturality of $\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}$ implies that the following diagram commutes:

$$\begin{tikzcd}
\mathcal{E}(\mathcal{E}(D)) \arrow[r, "\mathcal{E}\left(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_D\right)"] \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{\mathcal{E}(D)}"'] & \mathcal{E}(\mathcal{E}'(D)) \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{\mathcal{E}'(D)}"] \\
\mathcal{E}'(\mathcal{E}(D)) \arrow[r, "\mathcal{E}'\left(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_D\right)"] & \mathcal{E}'(\mathcal{E}'(D))
\end{tikzcd}$$

This definition (108) of monad morphism is implicit in [Bé67, pp. 39] (whose identification of monads as lax 2-functors out of the terminal category implies that their morphisms should be the corresponding lax natural transformations), first made explicit in [Mar66] and then in [Fr69][Pu70, p. 330][Str72, pp. 150][23], often in slight further generality. A transparent textbook account is in [BW85, §6.1], discussion in the context of monadic computations effects is in [Mog89a, Def. 4.0.11].

One readily checks[24] that the conditions (104) and (108) are equivalent under the translation (71); in particular the naturality of the transformation (107) is already implied by (104).

If we denote by Mnd(Type) the category whose objects are the monads on the type system and whose morphisms are monad transformations in the form (107), then their equivalence with (106) means that we have a faithful functor from monad transformations to functors between free modales:

$$\begin{array}{ccc}
\mathrm{Mnd}(\mathrm{Type}) & \longrightarrow & \mathrm{Type}/\mathrm{Cat} \\
\mathcal{E} & \longmapsto & \mathrm{Type}_{\mathcal{E}}.
\end{array}$$

---

[22] `hackage.haskell.org/package/transformers-0.5.6.2/docs/Control-Monad-Trans-Class.html#g:1`

[23] Beware that [Str72] says "transformation" for the 2-morphisms in the 2-category of monads, while we use it for the 1-morphisms, matching the completely standard terminology for the 1-morphisms of their underlying endofunctors and staying close to the established use of "monad transformers" (110).

[24] We are not aware of an explicit reference providing this equivalence; for the record we have spelled it out at: `ncatlab.org/nlab/show/monad+transformer#EquivalenceOfDefinitions`.

(This is known to experts but scarcely represented in the literature: The functor is alluded to in [Li69, Lem. 10.2] and only recently was discussed [AMcD23, Cor. 6.49] in detail but much more abstractly.)

For example, there is a *unique* transformation from the identity monad (the trivial effect) to any other monad $\mathcal{E}$, making the identity monad the initial object in the category of monads:

$$\exists!\, \mathtt{trans}^{\mathrm{Id} \to \mathcal{E}} : \mathrm{Id} \to \mathcal{E}, \quad \text{since} \quad \begin{tikzcd} D \arrow[r, equal] \arrow[d, equal] & D \arrow[d, "\mathrm{ret}^{\mathcal{E}}_D"] \\ D \arrow[r, "\mathrm{trans}^{\mathrm{Id}\to\mathcal{E}}_D", ":=\mathrm{ret}^{\mathcal{E}}_D"'] & \mathcal{E}(D)\,, \end{tikzcd} \qquad \begin{tikzcd} D \arrow[r, "\mathrm{ret}^{\mathcal{E}}_D"] \arrow[d, equal] & \mathcal{E}(D) \arrow[r, "\mathrm{ret}^{\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[dr, equal] & \mathcal{E} \circ \mathcal{E}(D) \arrow[d, "\mathrm{join}^{\mathcal{E}}_D"] \\ D \arrow[rr, "\mathrm{ret}^{\mathcal{E}}_D"] & & \mathcal{E}(D)\,. \end{tikzcd} \tag{109}$$

But in fact, [LHJ95, p. 339] and the functional programming/`Haskell`-community following them impose a further condition on monad transformers $\mathtt{trans}^{\mathcal{E} \to \mathcal{E}'}_D$, namely that they themselves arrange into the component maps of a pointed endofunctor

$$\mathrm{Id} \xrightarrow{\mathrm{trans}} (\text{-})' \;:\; \mathrm{Mnd} \to \mathrm{Mnd} \tag{110}$$

on the category of monads (made explicit in this form in [Wi22, p. 474]). This is tailored towards the application of *combining* monadic effects and hence regarding $\mathcal{E}'$ as behaving like the composition of $\mathcal{E}$ with another effect.

In addition to the covariant functor on free modales (106), a transformation between monads (107) *contra*variantly induces ([Fr69, Thm. 2], cf. [BW85, Thm. 6.3]) a functor between their general modales (91) by what we may recognize as the usual "extension of scalars"-formula from algebra:

$$\begin{tikzcd} \mathcal{E}' & & \mathcal{E} \arrow[ll, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}"'] \\[-10pt] \mathrm{Type}^{\mathcal{E}'} \arrow[rr] & & \mathrm{Type}^{\mathcal{E}} \end{tikzcd} \tag{111}$$

$$\begin{tikzcd} \mathcal{E}'(D_1) \arrow[r, "\mathcal{E}'(\phi)"] \arrow[d, "\rho'_1"] & \mathcal{E}'(D_2) \arrow[d, "\rho'_2"] \\ D_1 \arrow[r, "\phi"] & D_2 \end{tikzcd} \quad \longmapsto \quad \begin{tikzcd} \mathcal{E}(D_1) \arrow[r, "\mathcal{E}(\phi)"] \arrow[d, dashed, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_1}"] \arrow[dd, bend right=60, "\rho_1"'] & \mathcal{E}(D_2) \arrow[d, dashed, "\mathrm{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}"] \arrow[dd, bend left=60, "\rho_2"] \\ \mathcal{E}'(D_1) \arrow[r, "\mathcal{E}'(\phi)"] \arrow[d, "\rho'_1"] & \mathcal{E}'(D_2) \arrow[d, "\rho'_2"] \\ D_1 \arrow[r, "\phi"] & D_2 \end{tikzcd}$$

Composite effect monads. With computational side-effects encoded by monads $\mathcal{E}, \mathcal{E}'$, ..., one is bound to consider *combined effects* encoded by *composite monads*

$$\mathcal{E}' \circ \mathcal{E} \;:\; \mathrm{Type} \to \mathrm{Type}\,. \tag{112}$$

In order for the combined join-operation on the composite underlying functors to exist in an evident way, one needs a natural transformation between the two possible orders of composition

$$\mathtt{distr}^{\mathcal{E},\,\mathcal{E}'} : \mathcal{E} \circ \mathcal{E}' \longrightarrow \mathcal{E}' \circ \mathcal{E}\,, \tag{113}$$

because then the candidate composite join-operation is this:

$$\begin{tikzcd} & & \mathcal{E}' \circ \mathcal{E} \circ \mathcal{E} \arrow[dr, "\mathcal{E}'(\mathrm{join}^{\mathcal{E}}_{(-)})"] & \\ \mathcal{E}' \circ \mathcal{E} \circ \mathcal{E}' \circ \mathcal{E} \arrow[r, "\mathcal{E}'(\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{\mathcal{E}(-)})"'] \arrow[rrr, rounded corners, to path={|- ([yshift=2ex]\tikztotarget.north) -- (\tikztotarget)}, "\mathrm{join}^{\mathcal{E}' \circ \mathcal{E}}_{(-)}"] & \mathcal{E}' \circ \mathcal{E}' \circ \acute{\mathcal{E}}' \circ \mathcal{E} \arrow[ur, "\mathrm{join}^{\mathcal{E}'}_{\mathcal{E}\circ\mathcal{E}(-)}"] \arrow[dr, "\mathcal{E}' \circ \mathcal{E}'(\mathrm{join}^{\mathcal{E}}_{(-)})"'] & & \mathcal{E}' \circ \mathcal{E}\,, \\ & & \mathcal{E}' \circ \mathcal{E}' \circ \mathcal{E} \arrow[ur, "\mathrm{join}^{\mathcal{E}'}_{\mathcal{E}(-)}"'] & \end{tikzcd} \qquad \begin{tikzcd} & \mathcal{E}' \arrow[dr, "\mathcal{E}'(\mathrm{ret}^{\mathcal{E}}_{(-)})"] & \\ \mathrm{id} \arrow[ur, "\mathrm{ret}^{\mathcal{E}'}_{\mathcal{E}(-)}"] \arrow[rr, "\mathrm{ret}^{\mathcal{E}' \circ \mathcal{E}}_{(-)}" description] \arrow[dr, "\mathrm{ret}^{\mathcal{E}}_{(-)}"'] & & \mathcal{E}' \circ \mathcal{E}\,. \\ & \mathcal{E} \arrow[ur, "\mathrm{ret}^{\mathcal{E}'}_{(-)}"'] & \end{tikzcd} \tag{114}$$

For this construction to satisfy the monad axioms (72), the distributivity transformation (113) needs to make the following diagrams commute ([Be69, §1], review in [BW85, §9 2.1]):

$$\begin{tikzcd}
\mathcal{E} \arrow[rr, equal] \arrow[d, "\mathcal{E}\left(\mathrm{ret}^{\mathcal{E}'}_{(-)}\right)"'] & & \mathcal{E} \arrow[d, "\mathrm{ret}^{\mathcal{E}'}_{\mathcal{E}(-)}"] \\
\mathcal{E} \circ \mathcal{E}' \arrow[rr, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}"] & & \mathcal{E}' \circ \mathcal{E}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{E}' \arrow[rr, equal] \arrow[d, "\mathrm{ret}^{\mathcal{E}}_{\mathcal{E}'(-)}"'] & & \mathcal{E}' \arrow[d, "\mathcal{E}'\left(\mathrm{ret}^{\mathcal{E}}_{(-)}\right)"] \\
\mathcal{E} \circ \mathcal{E}' \arrow[rr, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}"] & & \mathcal{E}' \circ \mathcal{E}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{E} \circ \mathcal{E}' \circ \mathcal{E}' \arrow[r, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{\mathcal{E}'(-)}"] \arrow[d, "\mathcal{E}\left(\mathrm{join}^{\mathcal{E}'}_{(-)}\right)"'] & \mathcal{E}' \circ \mathcal{E} \circ \mathcal{E}' \arrow[r, "\mathcal{E}'\left(\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}\right)"] & \mathcal{E}' \circ \mathcal{E}' \circ \mathcal{E} \arrow[d, "\mathrm{join}^{\mathcal{E}'}_{\mathcal{E}(-)}"] \\
\mathcal{E} \circ \mathcal{E}' \arrow[rr, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}"] & & \mathcal{E}' \circ \mathcal{E}
\end{tikzcd} \tag{115}$$

$$\begin{tikzcd}
\mathcal{E} \circ \mathcal{E} \circ \mathcal{E}' \arrow[r, "\mathcal{E}\left(\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}\right)"] \arrow[d, "\mathrm{join}^{\mathcal{E},\mathcal{E}'}_{\mathcal{E}'(-)}"'] & \mathcal{E} \circ \mathcal{E}' \circ \mathcal{E} \arrow[r, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{\mathcal{E}(-)}"] & \mathcal{E}' \circ \mathcal{E} \circ \mathcal{E} \arrow[d, "\mathcal{E}'\left(\mathrm{join}^{\mathcal{E},\mathcal{E}'}_{(-)}\right)"] \\
\mathcal{E} \circ \mathcal{E}' \arrow[rr, "\mathtt{distr}^{\mathcal{E},\mathcal{E}'}_{(-)}"] & & \mathcal{E}' \circ \mathcal{E}
\end{tikzcd}$$

Computational contexts and co-monads on the type system. All of the above discussion of effect-monads 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* [UV08][POM13]. 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 $C(D)$ which is such that the context-part of the adjusted data is being transferred (extended) to followup programs:

$$\begin{tikzcd}
& \text{first program} & & \text{second program} & \\
C(D_1) \arrow[r, "\mathrm{prog}_{12}", "\substack{\text{output data}\\ \text{of type } D_2\\ \text{obtained in context of type } C(-)}"'] & D_2 & C(D_2) \arrow[r, "\mathrm{prog}_{23}", "\substack{\text{input data}\\ \text{of nominal type } D_2\\ \text{having context of type } C(-)}"'] & D_3 & {} \arrow[d, bend left, "\substack{\text{extend previous context}\\ \text{over second program}}"] \\
C(D_1) \arrow[r, "\mathtt{extend}^{\mathcal{E}}\mathrm{prog}_{12}", "\substack{\text{extend any previous}\\ C(-)\text{-context going forward}}"'] & C(D_2) & C(D_2) \arrow[r, "\mathrm{prog}_{23}"] & D_3 & {} \arrow[dd, bend left, "\text{compose}"] \\
& \text{C-composite program} & & & \\
C(D_1) \arrow[rrr, "\mathrm{prog}_{23} \circ \mathtt{extend}^{C}\mathrm{prog}_{12}", "\text{in shared } C(-)\text{-context}"'] & & & D_3 & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
C(D) \arrow[r, "\mathtt{obtain}^{C}_{D}", "\text{obtain plain data from } C(-)\text{-context}"'] & D \\
\mathcal{E}(D) \arrow[r, "\mathtt{extend}^{C}\mathtt{obtain}^{C}_{D}", "= \mathrm{id}_{\mathcal{E}(D)}"'] & \mathcal{E}(D)
\end{tikzcd} \tag{116}$$

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 (74):

$$\begin{tikzcd}
\mathrm{Type}' \arrow[r, bend right, "R"', "\text{right adjoint}"'] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, bend right, "L"', "\text{right adjoint}"] \arrow[loop right, "L \circ R =: C"] & \text{induced co-monad}
\end{tikzcd} \tag{117}$$

Examples of context comonads. Dualizing the example of the state monad (83) yields the **costate comonad** (or *store comonad*, cf. [Mi19][Uu21, 3, p. 14]):

$$\begin{array}{rcl}
W\mathrm{Store} \ : \mathrm{Type} & \longrightarrow & \mathrm{Type} \\
D & \longmapsto & W \times [W,\, D]
\end{array} \tag{118}$$

with operations

$$\begin{array}{lcl}
\mathtt{obtain}^{W\mathrm{Store}}_{D} & : & W \times [W, D] \to D \\
\mathtt{obtain}^{W\mathrm{Store}}_{D} & \equiv & (w, f) \mapsto f(w)
\end{array}
\qquad
\begin{array}{lcl}
\mathtt{extend}^{W\mathrm{Store}}D & : & (W \times [W, D] \to D) \mapsto (W \times [W, D]) \\
\mathtt{extend}^{W\mathrm{Store}}D & \equiv & ((w, f) \mapsto O(w, f)) \mapsto \big((w, f) \mapsto (w,\, O(\text{-}, f))\big)
\end{array} \tag{119}$$

which means that $W\mathrm{Store}(D)$ is the type of W-indexed supply ("storage") $f : W \to D$ of D-data equipped with an address $w : W$ of one such D-datum, which is the one that is `obtained` from such a computational context.

Similarly, dualizing the previous examples (82)(81) 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

(120)

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 §2. In this case their categories of (co)modales (93) are isomorphic (e.g. [MLM92, §V.8, Thm. 2]):

$$\underset{\text{adjoint (co)monads}}{\mathcal{E} \dashv \mathcal{C}} \qquad \overset{\text{have}}{\vdash} \qquad \begin{tikzcd}[column sep=small] \mathrm{Type}^{\mathcal{E}} \arrow[rr, leftrightarrow, "\sim", "\text{equivalent categories of modales}"'{yshift=20pt}] \arrow[dr, "U^{\mathcal{E}}"'] & & \mathrm{Type}^{\mathcal{C}} \arrow[dl, "U^{\mathcal{C}}"] \\ & \mathrm{Type} & \end{tikzcd} \tag{121}$$

Frobenius monads. Something special happens here when the underlying endo-functors in (121) 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 (98) and (117) from an "ambidextrous" adjunction, where the left and the right adjoint of a middle functor agree

$$\begin{tikzcd} & \mathcal{E} \arrow[dr, bend left] \arrow[dd, phantom, "\wr" description] & & \\ \text{Frobenius monad} & & \mathrm{Type} \arrow[r, shift left=10pt, "L \equiv R", "\text{ambidextrous adjunction}"{yshift=20pt}] \arrow[r, phantom, shift left=5pt, "\perp"] \arrow[r, phantom, shift right=5pt, "\perp"] \arrow[r, shift right=10pt, "R \equiv L"'] & \mathrm{Type}\,, \arrow[l, "C" description] \\ & \mathcal{C} \arrow[ur, bend right] & & \end{tikzcd} \tag{122}$$

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 (117): if type A carries Frobenius algebra structure, then the induced (Co)Reader (co)monad $A \otimes (\text{-})$ carries induced Frobenius monad structure.

Combined contextful and effectful programs. We have seen effectful programs typed as maps into monad types $\mathcal{E}(-)$ (67) and contextful programs typed as maps out of comonad types $C(-)$ (116). Of course, in general a program may be *both* effectful as well as context-dependent, in which case it should clearly be a map of the form

$$\mathrm{prog}_{12} \;\;:\;\; C(D_1) \longrightarrow \mathcal{E}(D_2)\,. \tag{123}$$

In order for such procedures to have a consistent composition, the context-comonad C needs to be compatible with the effect-monad $\mathcal{E}$ in the following way, known as a *distributivity law* for comonads over monads ([BVS93, Def. 3][25]). Namely, the order of application of the (co)monads must be interchangeable via a natural transformation

$$D \,:\, \mathrm{Type} \qquad \vdash \qquad \mathrm{distr}^{C,\mathcal{E}}_{D} \;:\; C(\mathcal{E}(D)) \longrightarrow \mathcal{E}(C(D)) \tag{124}$$

that make the following diagrams commute, not unlike the conditions on monad transformations (103):

$$\begin{tikzcd}
C(D) \arrow[r, equal] \arrow[d, "C(\mathrm{ret}^{\mathcal{E}}_{D})"'] & C(D) \arrow[d, "\mathrm{ret}^{\mathcal{E}}_{C(D)}"] \\
C(\mathcal{E}(D)) \arrow[r, "\mathrm{distr}^{C,\mathcal{E}}_{D}"] & \mathcal{E}(C(D))
\end{tikzcd}
\qquad
\begin{tikzcd}
C(\mathcal{E}(D)) \arrow[r, "\mathrm{distr}^{C,\mathcal{E}}_{D}"] \arrow[d, "\mathrm{obt}^{C}_{\mathcal{E}(D)}"'] & \mathcal{E}(C(D)) \arrow[d, "\mathcal{E}(\mathrm{obt}^{C}_{D})"] \\
\mathcal{E}(D) \arrow[r, equal] & \mathcal{E}(D)
\end{tikzcd}$$

$$\begin{tikzcd}
C\,(\mathcal{E}(\mathcal{E}(D))) \arrow[r, "\mathrm{distr}^{C,\mathcal{E}}_{\mathcal{E}(D)}"] \arrow[d, "C(\mathrm{join}^{\mathcal{E}}_{D})"'] & \mathcal{E}\,(C(\mathcal{E}(D))) \arrow[r, "\mathcal{E}(\mathrm{distr}^{C,\mathcal{E}}_{D})"] & \mathcal{E}\,(\mathcal{E}(C(D))) \arrow[d, "\mathrm{join}^{\mathcal{E}}_{C(D)}"] \\
C(\mathcal{E}(D)) \arrow[rr, "\mathrm{distr}^{C,\mathcal{E}}_{D}"] & & \mathcal{E}(C(D))
\end{tikzcd} \tag{125}$$

$$\begin{tikzcd}
C(\mathcal{E}(D)) \arrow[rr, "\mathrm{distr}^{C,\mathcal{E}}_{D}"] \arrow[d, "\mathrm{dupl}^{C}_{\mathcal{E}(D)}"'] & & \mathcal{E}(C(D)) \arrow[d, "\mathcal{E}(\mathrm{dupl}^{C}_{D})"] \\
C\,(C(\mathcal{E}(D))) \arrow[r, "C(\mathrm{distr}^{C,\mathcal{E}}_{D})"] & C\,(\mathcal{E}(C(D))) \arrow[r, "\mathrm{distr}^{C,\mathcal{E}}_{C(D)}"] & \mathcal{E}\,(C(C(D)))
\end{tikzcd}$$

With such distributivity structure, the C-context-dependent $\mathcal{E}$-effectful programs (123) have a consistent composition ([BVS93, Thm. 3][PW02, Prop. 7.4]) by combining the C-context extension (116) of the first with the $\mathcal{E}$-effect binding (67) of the second, concatenated via the distributivity transformation (124):

$$\begin{tikzcd}
C(D_1) \arrow[r, "\mathrm{prog}_{12}"] & \mathcal{E}(D_2)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
C(D_2) \arrow[r, "\mathrm{prog}_{23}"] & \mathcal{E}(D_3)
\end{tikzcd}$$

$$\begin{tikzcd}
C(D_1) \arrow[r, "\mathrm{dupl}^{C}_{D}"] \arrow[rr, bend left, "\texttt{extend}^{C}\mathrm{prog}_{12}"] \arrow[rrrrrr, bend right, "\mathrm{prog}_{12}\;\texttt{>=>}\;\mathrm{prog}_{23}"'] & C(C(D_1)) \arrow[r, "C(\mathrm{prog}_{12})"] & C(\mathcal{E}(D_2)) \arrow[r, "\mathrm{distr}^{C,\mathcal{E}}_{D_2}"] & \mathcal{E}(C(D_2)) \arrow[r, "\mathcal{E}(\mathrm{prog}_{23})"] \arrow[rr, bend left, "\texttt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(\mathcal{E}(D_3)) \arrow[r, "\mathrm{join}^{\mathcal{E}}_{D_3}"] & \mathcal{E}(D_3)
\end{tikzcd} \tag{126}$$

Notice that for $C = \mathrm{Id}$ or $\mathcal{E} = \mathrm{Id}$ the trivial (co)monad also the distributivity may be taken to be the identity and then this composition reduces to the Kleisli composition (69) of purely contextful- or purely effectful programs, whence we may use the same notation >=> also for this general case.

Literature 1.18 (Classical structures via Frobenius monads). The QuantumEnvironment (co)monad expressing quantum measurement effects which we derive in Prop. 2.35 (cf. Rem. 2.44 and p. 9) was originally considered for this purpose in [CPav08][CPaq08][CPP0909][CPV12], partial review in [HV19]. Its graphical formalization as part of the `zxCalculus`[26] (review in: [vWe][Co23]) originates in [CD08, §3][CD11, Def. 6.4][Ki08, §§2][Ki09, §4].

Literature 1.19 (Programming language for monadic effects). With a good categorical semantics in hand for effectful functional programs via monads (Lit. 1.17) one is left with finding a good syntax for neatly expressing such constructions

---

[25] Beware that [BVS93, Def. 3] refer to (124) as the *monad distributing over the comonad* instead of the other way around (therein following convention for the original discussion of monads distributing over monads in [Be69, §1]); but comparison with the eponymous case in arithmetic — $a \times \sum_i b_i \;\mapsto\; \sum_i a \times b_i$ — as well as with our main Ex. 2.56 makes our converse terminology more natural, which also coincides with the terminology used in [PW02, p. 138]. In any case, the formulas will always make unambiguously clear what is meant.

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

inside a given programming language (a “domain-specific embedded language”, Lit. 1.6). We review the traditional such syntax known as “`do`-notation” but highlight that — for conceptual clarity and for generalization to linear data types (Lit. 1.4) — this is better cast in `for`...`do`-form, which is what we use for our quantum pseudo-code in §3.

Traditional `do`-notation. The main example of an existing programming language with support for monadic effects is `Haskell`.[27] Here the (Kleisli-)composition of $\mathcal{E}$-effectful programs via effect-binding (67) is encoded by “`do`-notation” (due to [Lau93, §3.3], see [HHPW07, p. 25], and adopted in `Haskell` since v1.3,[28] for review see [BHM02, p. 70][Mi19, §20.3]). First of all, `do`-notation is suggestive syntax for the operation of effect-binding (67)

$$
\begin{array}{lcl}
\texttt{bind}^{\mathcal{E}}\mathrm{prog} & : & \mathcal{E}D \to \mathcal{E}D' \\
\texttt{bind}^{\mathcal{E}}\mathrm{prog} & \equiv & \texttt{E} \mapsto \left[\begin{array}{l} \texttt{do} \\ \quad d \leftarrow \texttt{E} \\ \quad \mathrm{prog}(d) \end{array}\right.
\end{array}
\tag{127}
$$

but thereby it furthermore provides a convenient means of expressing successive Kleisli-composition simply by successive “calling” of separate procedures, much in the style of “imperative” programming (which is thereby emulated into functional programming, Lit. 1.16):

Composite Kleisli morphism

$$\begin{tikzcd}
D_1 \arrow[r, no head, "\mathrm{id}(D_1 \xrightarrow{\texttt{this}} \mathcal{E}(D_2))"] & {} \arrow[dl, rounded corners] \\
\mathcal{E}(D_2) \arrow[r, no head, "\texttt{bind}^{\mathcal{E}}(D_2 \xrightarrow{\texttt{that}} \mathcal{E}(D_3))"] & {} \arrow[dl, rounded corners] \\
\mathcal{E}(D_3) \arrow[r, equal, "\texttt{bind}^{\mathcal{E}}(\texttt{return}^{\mathcal{E}}_{D_3})"] & \mathcal{E}(D_3)
\end{tikzcd} \tag{128}$$

Corresponding do-notation

```
d1 -> do
  d2 <- this d1
  d3 <- that d2
  return d3
```

(For the moment we closely stick to `Haskell` typewriter-style typesetting on the right, just for ease of comparison, but in §3 we use more fonts to better guide the eye.)

This notation is particularly suggestive due to the further convention that the variable names may be suppressed for functions with trivial in- or out-put (i.e. of unit type $*$, such for programs whose only purpose is write to a log as in (81)) besides their $\mathcal{E}$-effect:

Composite Kleisli morphism

$$\begin{tikzcd}
* \arrow[r, no head, "\mathrm{id}(* \xrightarrow{\texttt{this}} \mathcal{E}(*))"] & {} \arrow[dl, rounded corners] \\
\mathcal{E}(*) \arrow[r, no head, "\texttt{bind}^{\mathcal{E}}(* \xrightarrow{\texttt{that}} \mathcal{E}(*))"] & {} \arrow[dl, rounded corners] \\
\mathcal{E}(*) \arrow[r, equal, "\texttt{bind}^{\mathcal{E}}(\texttt{return}^{\mathcal{E}}_{*})"] & \mathcal{E}(*)
\end{tikzcd}$$

Corresponding do-notation

```
do
  this
  that
return
```

Here it is manifest how the outer `do...return`-block syntax expresses the consecutive Kleisli-composition of any number effectful procedures.

On top of that, the “`<-`”-syntax is meant to be suggestive of *reading out* a value from an effectful datum. This imagery is accurate in case of the State-monad (83) (particularly in its incarnation as the `IO`-monad [PW93] modelling actual machine reading from an input device such as a keyboard and machine writing to an output device such as a file). To make this explicit, consider the following stateful programs for reading/writing the state of a global variable of type W:

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
\texttt{read}_W & : & W\mathrm{State}(W) & \texttt{write}_W & : & W \to W\mathrm{State}(*) \\
\texttt{read}_W & \equiv & w \mapsto (w, w) & \texttt{write}_W & \equiv & w \mapsto (w' \mapsto w)
\end{array}
\tag{129}
$$

---

[27] `Haskell` landing page: `www.haskell.org`

[28] `www.haskell.org/definition/from12to13.html#do`

From these, all other stateful operations may be composed via `do`-notation. For instance, the operation which increments a global integer variable

$$\begin{aligned} \texttt{inc} &: \mathbb{Z}\mathrm{State}(\ast) \\ \texttt{inc} &\equiv n \mapsto n+1 \end{aligned}$$

may be coded as follows, cf. (85), and the example in [BHM02, p. 68 & 71]:

Composite Kleisli morphism

$$\begin{tikzcd}
\ast \arrow[d, "\mathrm{id}(\ast \xrightarrow{\texttt{read}} \mathbb{Z}\mathrm{State}(\mathbb{Z}))"] \\
\mathbb{Z}\mathrm{State}(\mathbb{Z}) \arrow[d, "\mathbb{Z}\mathrm{State}(\mathbb{Z} \xrightarrow{+1} \mathbb{Z})"] \\
\mathbb{Z}\mathrm{State}(\mathbb{Z}) \arrow[d, "\texttt{bind}(\mathbb{Z} \xrightarrow{\texttt{write}} \mathbb{Z}\mathrm{State}(\ast))"] \\
\mathbb{Z}\mathrm{State}(\ast) \arrow[r, equal, "\texttt{bind}(\texttt{return}^{\mathbb{Z}\mathrm{State}}_{\ast})"] & \mathbb{Z}\mathrm{State}(\ast)
\end{tikzcd}$$

Corresponding do-notation

```
do
    n <- read
    write n + 1
  return
```

In this case it is nicely suggestive that the line "`n <- read`" instructs to read out the given state and to bind its value to the variable `n`. However, already for similar effect monads such as the list monad ([Wa90, 2.1][Mi19, pp 304])

$$\begin{array}{rcl} \mathrm{List}\ : \mathrm{Type} & \longrightarrow & \mathrm{Type} \\ D & \longmapsto & \coprod_{n:\mathbb{N}} D^{\times n} \end{array} \qquad \begin{array}{ccc} D & \xrightarrow{\mathrm{ret}^{\mathrm{List}}_D} & \mathrm{List}(D) \\ d & \mapsto & (d) \end{array} \qquad \begin{array}{ccc} \mathrm{List}(\mathrm{List}(D)) & \xrightarrow{\mathrm{join}^{\mathrm{List}}_D} & \mathrm{List}(D) \\ \left(\begin{array}{l} (d_{11}, \cdots, d_{1n_1}), \\ \quad \vdots \\ (d_{k1}, \cdots, d_{1n_k}) \end{array}\right) & \mapsto & (d_{11}, \cdots, d_{1n_1}, \cdots, d_{k1}, \cdots, d_{kn_k}) \end{array} \tag{130}$$

the idea of Kleisli composition as being about "reading out" intermediate variables is a little inaccurate. For example, the operation of incrementing all entries in a list of integers is coded in `do`-notation as follows:

Composite Kleisli morphism

$$\begin{tikzcd}
\ast \arrow[d, "\mathrm{id}(\ast \xrightarrow{\texttt{MyList}} \mathrm{List}(\mathbb{Z}))"] \\
\mathrm{List}(\mathbb{Z}) \arrow[d, "\mathrm{List}(\mathbb{Z} \xrightarrow{+1} \mathbb{Z})"] \\
\mathrm{List}(\mathbb{Z}) \arrow[r, equal, "\texttt{bind}^{\mathrm{List}}(\texttt{return}^{\mathrm{List}}_{\mathbb{Z}})"] & \mathrm{List}(\mathbb{Z})
\end{tikzcd} \tag{131}$$

Corresponding do-notation

```
do
  n <- MyList
return n + 1
```

Here the code on the right nicely evokes the idea that we are "reading out" an element from the list and `return`ing its increment — but it leaves linguistically implicit the crucial fact that this process is to be applied *for all* elements of the list, and that the results be re-compiled into an output list: Instead of just "`do this`", the natural-language rendering of the above list algorithm would be more like "`do this for any element`".

For-Do-Notation. Indeed, we may observe in generality that it is misleading to think of effect-composition as being about "reading out" data elements: Rather, Kleisli morphisms, in their nature as $(U^{\mathcal{E}} \dashv F^{\mathcal{E}})$-adjuncts (98)(75) of modale homomorphisms out of *free* modales

$$\begin{tikzcd}
\underset{\textbf{free modale}}{\mathcal{E}(D_1)} \arrow[loop above, "\mathcal{E}"] \arrow[r, "\overset{\textbf{$\mathcal{E}$-modale homomorphism}}{\widetilde{\mathrm{prog}}}"] & \mathcal{E}(D_2) \arrow[loop above, "\mathcal{E}"] & \leftrightarrow & \underset{\textbf{generating data}}{D_1} \arrow[r, "\overset{\textbf{Kleisli map}}{\mathrm{prog}}"] & \mathcal{E}(D_2)
\end{tikzcd}$$

are about acting on freely *generated data* types $\mathcal{E}(D)$ by declaring how to operate *on generators* $d : D$, hence about what to do *for* a given generator.

Therefore, we may argue that the program-linguistically more evocative rendering of what is going on in monadic effect-binding operation is a slight enrichment of the traditional `do`-notation to a ”`for`...`do`”-block, as follows:

Monadic effect binding

$$\mathcal{E}(D) \xrightarrow{\texttt{bind}^{\mathcal{E}}(\text{prog})} \mathcal{E}(D') \qquad (132)$$

Corresponding for-do-notation

```
E ↦ | for d in E
    |
    |  do prog d
```

(Notice that in imperative languages the `for`...`do`-syntax is traditionally used to code loops, but in the functional languages that we are concerned with such loops are instead coded by recursion, so that the `for`...`do`-syntax does remain free to be used for the purpose of effect binding.)

In this notation, the generic example (128) is rendered into code as follows:

Composite Kleisli morphism

$$\begin{tikzcd}
D_1 \arrow[r, "\mathrm{id}(D_1 \xrightarrow{\texttt{this}} \mathcal{E}(D_2))"] & \mathcal{E}(D_2) \arrow[r, "\texttt{bind}^{\mathcal{E}}(D_2 \xrightarrow{\texttt{that}} \mathcal{E}(D_3))"] & \mathcal{E}(D_3) \arrow[r, equal, "\texttt{bind}^{\mathcal{E}}(\texttt{return}^{\mathcal{E}}_{D_3})"] & \mathcal{E}(D_3)
\end{tikzcd}$$

Corresponding for-do-notation

```
prog  :  $D_1 \to \mathcal{E}(D_3)$

prog  ≡  | for d2 in this d1
         |
         |  do | for d3 in that d2
         |     |
         |     |  do return d3
```

This may be notationally less concise than (128) but in its close relation to natural language rendering of the computational process it lends itself to the formulation of transparent pseudocode such as we consider in §3, especially when it comes to operations on linear types, cf. (251).

For instance, in this `for`...`do`-notation the previous example (131) of entry-wise increments in a list now reads as follows, neatly indicative of how the incremenet is applied *for* every element *n* found *in* the given list *L*:

Composite Kleisli morphism

$$\mathrm{List}(\mathbb{Z}) \xrightarrow{\texttt{bind}\left(\mathbb{Z} \xrightarrow{+1} \mathbb{Z} \xrightarrow{\texttt{return}} \mathrm{List}(\mathbb{Z})\right)} \mathrm{List}(\mathbb{Z})$$

Corresponding for-do-notation

```
inc  :  $\mathrm{List}(\mathbb{Z}) \to \mathrm{List}(\mathbb{Z})$

inc  ≡  L -> | for n in L
             |
             |  do   return n + 1
```

1.4 Monoidal categories

Literature 1.20 (**Monoidal categories of quantum types).** One of the key distinctions between classical and quantum types (Lit. 1.4) is the nature of their logical conjunction, reflected in a *monoidal structure* ([EK66, §II.1][ML71/97, §VII] [Bor94b, §6.1]) on the categories that they form.

Purely classical types should form a (locally) *cartesian* closed category, while purely quantum types should form a symmetric monoidal closed category which is non-cartesian (24) to admit a good supply of dualizable (finite-dimensional) types:

Dualizable/Finite-dimensional linear types. Somewhat in contrast to quantum theory in general, the focus of quantum computation/information-theory is on quantum systems with *finite-dimensional* (Hilbert-)spaces $\mathcal{H}$ of quantum states (Lit. 1.1), whose characteristic property is that they are the dual spaces $(\mathcal{H}^*)^*$ of their own dual spaces.

Abstractly, the characterization of finite-dimensionality of an object $\mathcal{H}$ in a symmetric monoidal category is its *strong dualizability* [DP84, §1] (indeed originally called "finite objects" in [Par76, p. 113]), given equivalently [DP84, Thm. 1.3] by the existence of an object $\mathcal{H}^*$ (to be called its *dual object*) and of morphisms

$$\mathbb{1} \xrightarrow{\;\mathrm{cev}_{\mathcal{H}}\;} \mathcal{H} \otimes \mathcal{H}^* , \qquad \mathcal{H}^* \otimes \mathcal{H} \xrightarrow{\;\mathrm{ev}_{\mathcal{H}}\;} \mathbb{1} \tag{133}$$

such that the following diagrams commute:

$$\begin{tikzcd}
\mathcal{H} \arrow[r, "l_{\mathcal{H}}", "\sim"'] \arrow[rrrr, bend right, "\mathrm{id}"] & \mathbb{1} \otimes \mathcal{H} \arrow[r, "\mathrm{cev}_{\mathcal{H}} \otimes \mathrm{id}_{\mathcal{H}}"] & \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \arrow[r, "\mathrm{id}_A \otimes \mathrm{ev}_{\mathcal{H}}"] & \mathcal{H} \otimes \mathbb{1} \arrow[r, "r^{-1}_{\mathcal{H}}", "\sim"'] & \mathcal{H}
\\
\mathcal{H}^* \arrow[r, "r_{\mathcal{H}^*}", "\sim"'] \arrow[rrrr, bend right, "\mathrm{id}"] & \mathcal{H}^* \otimes \mathbb{1} \arrow[r, "\mathrm{id}_{\mathcal{H}} \otimes \mathrm{cev}_{\mathcal{H}}"] & \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\mathrm{ev}_{\mathcal{H}} \otimes \mathrm{id}_A"] & \mathbb{1} \otimes \mathcal{H}^* \arrow[r, "l^{-1}_{\mathcal{H}^*}", "\sim"'] & \mathcal{H}^*
\end{tikzcd}$$

(134)

This implies[29] that the tensor product functors with these objects are adjoint to each other (74) as

$$(\text{-}) \otimes \mathcal{H} \;\dashv\; (\text{-}) \otimes \mathcal{H}^* \tag{135}$$

with adjunction counit given by the evaluation map. By uniqueness of adjoints this means that when the ambient category is *closed* monoidal (as it is in all our applications) with internal hom $(\text{-}) \multimap (\text{-})$ then

$$(\text{-}) \otimes \mathcal{H}^* \;\simeq\; \mathcal{H} \multimap (\text{-}) \tag{136}$$

and hence in particular that

$$\mathcal{H}^* \;\simeq\; \mathbb{1} \otimes \mathcal{H} \;\simeq\; \mathcal{H} \multimap \mathbb{1} \,. \tag{137}$$

But by symmetry, the conditions (134) imply that $\mathcal{H} \simeq (\mathcal{H}^*)^*$ is the dual of its dual object, to that this adjunction is actually ambidextrous, in that

$$(\text{-}) \otimes \mathcal{H} \;\dashv\; (\text{-}) \otimes \mathcal{H}^* \;\dashv\; (\text{-}) \otimes \mathcal{H} \,. \tag{138}$$

Categories of internal modules. Sometimes it is useful to produce new categories of linear types from given ones by *internal algebra* (eg. [Boa95]): If $(C, \otimes, \mathbb{1})$ is a bicomplete symmetric monoidal closed category [EK66, §III], then for

$$A \in \mathrm{Mon}(C, \otimes, \mathbb{1}) \tag{139}$$

an internal *monoid object* [ML71/97, VII.3], i.e. an object $A \in C$ equipped with morphisms

$$\mathbb{1} \xrightarrow[\text{unit element}]{\;1_A\;} A \,, \qquad A \otimes A \xrightarrow[\text{product operation}]{\;(\text{-})\cdot(\text{-})\;} A \tag{140}$$

in C, making the following diagrams commute:

$$\begin{tikzcd}
& A \otimes A \arrow[dr, "(\text{-})\cdot(\text{-})"] \arrow[d, phantom, "\text{left unitality}"] & \\
\mathbb{1} \otimes A \arrow[ur, "1_A \otimes (\text{-})"] \arrow[rr, "\sim"'] & {} & A \,,
\end{tikzcd}
\quad
\begin{tikzcd}
& A \otimes A \arrow[dr, "(\text{-})\cdot(\text{-})"] \arrow[d, phantom, "\text{right unitality}"] & \\
A \otimes \mathbb{1} \arrow[ur, "(\text{-}) \otimes 1_A"] \arrow[rr, "\sim"'] & {} & A \,,
\end{tikzcd}
\quad
\begin{tikzcd}
A \otimes A \otimes A \arrow[r, "(\text{-})\cdot(\text{-})\otimes(\text{-})"] \arrow[d, "(\text{-})\otimes(\text{-})\cdot(\text{-})"] \arrow[dr, phantom, "\text{associativity}"] & A \otimes A \arrow[d, "(\text{-})\cdot(\text{-})"] \\
A \otimes A \arrow[r, "(\text{-})\cdot(\text{-})"] & A
\end{tikzcd}$$

(141)

---

[29] Beware that for $\mathcal{H}$ to be strong dualizable it is *not sufficient* that $(\text{-}) \otimes \mathcal{H}$ be a left adjoint. But an evaluation-type map on $\mathcal{H}$ does exhibit a strong duality iff it induces the counit of such an adjunction, this is [DP84, Thm. 1.3 (b) & (c)].

then its internal modules [ML71/97, VII.4], being objects $N \in C$ equipped with an action morphism in C

$$
A \otimes N \xrightarrow[\text{left action}]{\rho} N \quad \text{such that} \quad
\begin{tikzcd}
 & A \otimes N \arrow[dr, "\rho"] & \\
\mathbb{1} \otimes N \arrow[ur, "1_A \otimes (-)"] \arrow[rr, "\sim"] \arrow[rr, phantom, shift left=4ex, "\text{unitality}"] & & N
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
 & A \otimes (A \otimes N) \arrow[r, "(-)\otimes\rho"] & A \otimes N \arrow[dd, "\rho"] \\
(A \otimes A) \otimes N \arrow[ur, "\sim"] \arrow[d, "(-)\cdot(-)\otimes(-)"'] & \text{action property} & \\
A \otimes N \arrow[rr, "\rho"] & & N\,.
\end{tikzcd}
\tag{142}
$$

form a category, to be denoted,

$$
(\mathrm{Mod}_A,\, \otimes_A,\, A) \;\equiv\; \mathrm{Mod}_A(C, \otimes, \mathbb{1}) \tag{143}
$$

is itself

(i) bicomplete, where the forgetful functor $U \,:\, \mathrm{Mod}_A \to C$ creates both limits and colimits [Mar09, Lem. 1.2.14], in particular:

$$
U \circ \varinjlim(\text{-}) \;\simeq\; \varinjlim(U \circ \text{-})\,, \qquad U \circ \varprojlim(\text{-}) \;\simeq\; \varprojlim(U \circ \text{-})\,, \tag{144}
$$

(ii) symmetric monoidal closed [HSS00, Lem. 2.2.2 & 2.2.8][Mar09, Lem. 1.2.15-17][Bra14, Prop. 4.1.10], with tensor unit A and tensor product the evident coequalizer:

$$
N, N' \,:\, \mathrm{Mod}_A \qquad \vdash \qquad N \otimes A \otimes N' \rightrightarrows N \otimes N' \xrightarrow{\text{coeq}} N \otimes_A N'\,. \tag{145}
$$

For **example**:

(1.) In the most fundamental case, the ambient monoidal category $C \,\equiv\, \mathrm{Mod}_{\mathbb{Z}} \,\equiv\, \mathrm{Ab}$ is that of abelian groups, whose internal monoid objects (139) are equivalently rings, whose module objects are the ordinary modules:

$$
\mathrm{Rng} \;\simeq\; \mathrm{Mon}(\mathrm{Ab})\,.
$$

(2.) In slight variation, if one instead considers C to be the category set-indexed abelian groups equipped with the "external" tensor product and with the base ring regarded now as parameterized over the singleton, then its internal modules are the set-indexed modules which serve as quantum semantics in §2.1, see Rem. 2.5 there.

(3.) In further generalization along these lines, if the ambient monoidal category is that of flat real vector bundles over $\mathbb{Z}_2$-action groupoids and the internal monoid is the "Real complex numbers" , then its internal monoids are (discussed in [SS23-QR]) Atiyah's Real vector bundles (with capital "*R*") over, in this case, discrete base Real spaces. These turn out to play a profound role in the typing of Hermitian inner product structure and hence of (finite-dimensional) Hilbert spaces (26), as discussed in [SS23-QR].

1.5 Parameterized spectra

Literature 1.21 (Parameterized stable homotopy theory, Tangent ∞-toposes & Twisted cohomology). The language of `LHoTT` (Lit. 1.8) syntactically captures the following striking confluence of fundamental structures in algebraic topology and homotopy theory:

The dichotomy between spaces and motives. One may observe that the following two fundamental types of 1-categories (cf. 1.4):

- **(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:

$$R \in E_\infty\mathrm{Ring}(\mathrm{Spctr}) \qquad \vdash \qquad T^R\mathrm{Grpd}_\infty \;:\equiv\; \int_{W \in \mathrm{Grpd}_\infty} \mathrm{Mod}_R^W \;\in\; \mathrm{Topos}_\infty\,. \tag{146}$$

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 (147). We may thus think of (146) 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).

Infinitesimal cohesion and classicality. To pinpoint the nature of this logical context, notice that there is a canonical inclusion of Grpd_∞ into its tangent ∞-topos (146) 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"

$$\begin{tikzcd}[row sep=large]
\textbf{R-linear tangent ∞-topos} & T^R\mathrm{Grpd}_\infty \arrow[loop above, "\natural"', "\textbf{classical modality}"] \arrow[d, shift right=3ex, "p"'] \arrow[d, shift left=3ex, "p"] & = \int_{\mathcal{X}} R\mathrm{Mod}^{\mathcal{X}} & \textbf{flat ∞-bundles of R-module spectra} \\
\textbf{classical base ∞-topos} & \mathrm{Grpd}_\infty \arrow[u, hook, "0" description, "\dashv\;\;\dashv"]
\end{tikzcd} \tag{147}$$

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 (147) 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 cohesive modalities $\flat$ and $\int$ unify into a single ambidextrous modality as shown in (147), now to be denoted"$\natural$" (following [RFL21]), which we may think of as retaining the underlying classical aspect of types while discarding their infinitesimal/microscopic (quantum) aspects, see Prop. 2.7 for more.

Flat vector bundles and Indexed vector spaces. Specifically when $R = H\mathbb{K}$ is the Eilenberg-MacLane spectrum over a ring or even a field $\mathbb{K}$, then there is an equivalence ([Rob87][ScSh03, Thm. 5.1.6]) between the homotopy theory of $H\mathbb{K}$-module spectra and that of $\mathbb{K}$-chain complexes, hence between that of W-parameterized $H\mathbb{K}$-module spectral and that of *flat ∞-vector bundles* over W, also known as *∞-local systems* over W (see [SS23-EoS, §3.1] for more):

$$\overset{\substack{\textbf{parameterized}\\ H\mathbb{K}\textbf{-module spectra}}}{\mathrm{Mod}^{W}_{H\mathbb{K}}} \quad\simeq\quad \overset{\substack{\infty\textbf{-local systems of}\\ \textbf{chain complexes}}}{\mathrm{Ch}^{W}_{\mathbb{K}}}$$

and the *hearts* (Rem. 1.22) of these stable ∞-categories are the 1-categories of ordinary flat vector bundles hence of ordinary local systems of vector spaces:

$$\overset{\textbf{Vector spaces}}{\mathrm{Mod}_{\mathbb{K}}} \;\overset{\text{are}}{\simeq}\; \overset{\textbf{the heart}}{\heartsuit(\mathrm{Mod}_{H\mathbb{K}})} \;\overset{\text{of}}{\longhookrightarrow}\; \overset{H\mathbb{K}\textbf{-module spectra}}{\mathrm{Mod}_{H\mathbb{K}}} \qquad\qquad \overset{\substack{\textbf{Flat}\\ \textbf{vector bundles}}}{\mathrm{Mod}^{W}_{\mathbb{K}}} \;\overset{\text{are}}{\simeq}\; \overset{\textbf{the heart}}{\heartsuit(\mathrm{Mod}^{W}_{H\mathbb{K}})} \;\overset{\text{of}}{\longhookrightarrow}\; \overset{\substack{\textbf{parameterized}\\ H\mathbb{K}\textbf{-module spectra}}}{\mathrm{Mod}^{W}_{H\mathbb{K}}}$$

Over $W : \mathrm{Set} \subset \mathrm{Grpd}$ these are plain vector bundles over the discrete spaces W, hence W-indexed vector spaces, whence their Grothendieck construction is the free coproduct completion $\mathrm{Fam}_{\mathbb{K}}$ of vector spaces providing the categorical semantics of (Proto-)`Quipper` (Lit. 1.5) and the 0-sector of `LHoTT`, which we discuss in detail in §2.1:

$$\begin{tikzcd}
\substack{\textbf{Categorical semantics of}\\ \textbf{(Proto-)}\texttt{Quipper}\ \textbf{\&}\\ \textbf{0-sector of}\ \texttt{LHoTT}} & \substack{\textbf{Categorical semantics of}\\ \textbf{heart-sector of}\ \texttt{LHoTT}\\ \textbf{including Hermitian spaces}} & \substack{\textbf{Categorical semantics}\\ \textbf{of}\ \texttt{LHoTT}\ \textbf{including}\\ \textbf{topological effects}} \\
\mathrm{Fam}_{\mathbb{K}} \arrow[r, hook] & \mathrm{Loc}_{\mathbb{K}} \arrow[r, hook] & T^{H\mathbb{K}}\mathrm{Grpd}_{\infty} \\
\underset{W : \mathrm{Set}}{\int} \mathrm{Mod}^{W}_{\mathbb{K}} \arrow[r, hook] & \underset{W : \mathrm{Grpd}}{\int} \mathrm{Mod}^{W}_{\mathbb{K}} \arrow[r, hook'] & \underset{W : \mathrm{Grpd}_{\infty}}{\int} \mathrm{Mod}^{W}_{H\mathbb{K}}
\end{tikzcd} \tag{148}$$

In the middle, we are showing an intermediate ground which turns out to be useful for typing Hermitian structure on quantum types and hence captures the probabilistic aspect of quantum theory (Lit. 1.12):

Equivariance by homotopy type-dependency. For G a group, a spectrum parameterized over its delooping (its 1st Eilenberg-Maclane space) $\mathbf{B}G$ is equivalently a G-action on the underlying spectrum (also known as a "naïvely G-equivariant spectrum"). Generally, the slice over $\mathbf{B}G$, hence the types *dependent on* variables in context $\mathbf{B}G$ are types equipped with a G-action (see [SS21-EqB, Prop. 0.2.1][ss20-Orb, §2.2]):

<table>
<tr><th colspan="2">Equivariance by dependency on delooping</th></tr>
<tr><th>Syntax</th><th>Semantics</th></tr>
<tr><td>$\vdash \quad \mathrm{pt} : \mathbf{B}G$
$\vdash \quad \mathrm{Id}_{\mathbf{B}G}(\mathrm{pt}, \mathrm{pt}) \;\simeq\; G$</td><td>$$\begin{tikzcd}
\textbf{group}\ G \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \ast \arrow[d, "\vdash\mathrm{pt}"] \\
\ast \arrow[r, "\vdash\mathrm{pt}"] & \mathbf{B}G\ \textbf{delooping}
\end{tikzcd}$$</td></tr>
<tr><td>$\mathrm{pt} \,:\, \mathbf{B}G \quad \vdash \quad E_{\mathrm{pt}} : \mathrm{Type}$</td><td>$$\begin{tikzcd}
G\textbf{-action}\ E_{\mathrm{pt}} \arrow[loop above, "G"] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(pb)}"] & \overset{\substack{\textbf{homotopy quotient/}\\ \textbf{Borel construction}}}{E_{\mathrm{pt}} /\!\!/ G} \arrow[d] \\
\ast \arrow[r, "\vdash\mathrm{pt}"] & \mathbf{B}G
\end{tikzcd}$$</td></tr>
</table>

(149)

Twisted cohomology. Interestingly, the hom-spaces in the R-tangent ∞-topos (146) are sections of R-module bundles $\tau_{\mathcal{X}}$, which means [ABGHR14][FSS23, Prop. 3.5][ss20-Orb, p. 6] that their connected components form the *$\tau_{\mathcal{X}}$-twisted R-cohomology* $R^{\tau}(\mathcal{X})$ of $\mathcal{X}$ [MaSi06, §22.11]:

$$\left.\begin{array}{l} \mathcal{X} \,\in\, \mathrm{Grpd}_{\infty} \\ R \,\in\, E_{\infty}\mathrm{Rng}(\mathrm{Spctr}) \end{array}\right\} \quad\vdash\quad \mathrm{Maps}(0_{\mathcal{X}},\, R/\!\!/\mathrm{GL}_1(R)) \;=\; \left\{ \begin{tikzcd} & R/\!\!/\mathrm{GL}_1(R) \arrow[d, two heads] \\ \mathscr{X} \arrow[ur, dashed, "\text{cocycle in } R^{\tau}(\mathcal{X})"] \arrow[r, dashed, "\tau_{\mathcal{X}}", "\text{twist}"'] & B\mathrm{GL}_1(R) \end{tikzcd} \right\}. \tag{150}$$

This already suggests [Sch14b] 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. 1.23).

Remark 1.22 (**0-sector and Heart-sector of `LHoTT`).**
(i) By the *0-sector* of `LHoTT` (Lit. 1.8) 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]).
(ii) The *heart* of the t-structure (formed by the spectra whose homotopy groups are concentrated in degree 0) reflects the intended 0-sector of the given stable homotopy theory. Hence by the 0-sector of `LHoTT` we mean those types which are in the heart and whose *underlying* purely classical type is 0-truncated.

Literature 1.23 (**Topological quantum materials and Topological K-theory).** For extensive background and referencing see [SS23b].

2 Quantum Effects

We show that a system of basic (co)monads which is canonically *defineable* (via admissibke inference rule) in any dependent linear homotopy type theory which satisfies the Motivic Yoga (Def. 2.18) 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 (Q) (turning bits into q-bits, etc.), these effects notably include quantum measurement (○) and conditional quantum state preparation (☆), which turn out to correspond to Coecke et al.'s "classical structures" Frobenius monad.

2.1 Quantum Semantics

We lay out a concrete example (Def. 2.1 below) of a category that interprets the 0-sector (Rem. 1.22) of `LHoTT` relevant for expressing quantum circuits (in §2.4). Category-theoretically this example is elementary and standard (going back to [Bé85, §3.3][HT95, pp. 281]), but it is important in applications, e.g. as the established model for Proto-`Quipper` (Lit. 1.5, where it appears as [RS18, Def. 3.3] for the case that their fiber category $\overline{M}$ is the category $\mathrm{Mod}_{\mathbb{K}}$ of $\mathbb{K}$-vector bundles, essentially the "quantum sets" of [Ko20][KLM21, §2]). Here we highlight previously underappreciated aspects of this model (all shared by its homotopy-theoretic generalizations in [SS23-EoS]):

- its doubly closed monoidal structure (Prop. 2.3),
- its doubly strong monadic reflections (Prop. 2.7),
- its quantization/exponential modality (Prop. 2.9),
- its support of 6-operations motivic yoga (Prop. 2.19),

which make the model interpret an expressive modal/monadic/effectful quantum language, `QS`, in §3.

Definition 2.1 (**Category of linear bundle types).**
For the purpose of this section, we write "Type" for the category equivalently described as follows
(cf. [SS23-EoS], where this category is denoted "$\mathrm{Fam}_{\mathbb{K}}$"):

- Type is the free coproduct completion of $\mathrm{Mod}_{\mathbb{K}}$,
- Type is the category of *indexed sets* of $\mathbb{K}$-vector spaces,
- Type is the category of 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 $\mathrm{Mod}_{\mathbb{K}}^{W} \equiv \mathrm{Func}(W,\, \mathrm{Mod}_{\mathbb{K}})$ of W-indexed vector spaces (vector bundles over W):

Syntax	Semantics	
Types	**Category**	**Morphisms**
ClaType classical types	Set sets	$W \xrightarrow{\;f\;} W'$ maps
QuType linear types	$\mathrm{Mod}_{\mathbb{K}}$ vector spaces	$\mathcal{H} \xrightarrow{\;\phi\;} \mathcal{H}'$ linear maps
QuType_{W} W-dependent linear types	$\mathrm{Mod}_{\mathbb{K}}^{W}$ W-indexed vector space	$\begin{tikzcd} \begin{bmatrix} \mathcal{H}_{\bullet} \\ \downarrow \\ W \end{bmatrix} \arrow[r, "\phi_{\bullet}"] \arrow[r, phantom, shift right=3, "="] & \begin{bmatrix} \mathcal{H}'_{\bullet} \\ \downarrow \\ W \end{bmatrix} \end{tikzcd}$ W-indexed linear maps
Type linear bundle types	$\int_{W:\mathrm{Set}} \mathrm{Mod}_{\mathbb{K}}^{W}$ 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

(151)

When describing their linear fiber types concretely, we also denote linear bundle types and their hom-sets as follows (the bottom lines exhibiting the type-theoretic syntax, see Rem. 2.4):

$$\begin{aligned} \begin{bmatrix} \mathcal{H}_{\bullet} \\ \twoheaddownarrow \\ W \end{bmatrix} \equiv \begin{bmatrix} \mathcal{H}_{w} \\ \twoheaddownarrow \\ (w : W) \end{bmatrix} \qquad & \mathrm{Hom}\left(\begin{bmatrix} \mathcal{H}_{\bullet} \\ \downarrow \\ W \end{bmatrix}, \begin{bmatrix} \mathcal{H}'_{\bullet} \\ \downarrow \\ W' \end{bmatrix} \right) \simeq \big(f : \mathrm{Hom}(W,\, W')\big) \times \prod_{w} \mathrm{Hom}\big(\mathcal{H}_{w},\, \mathcal{H}'_{f(w)}\big) \\ \equiv\; (w : W) \times (\mathcal{H}_{w} : \mathrm{QuType}) \qquad & \qquad\qquad\qquad\qquad\quad \equiv \big(f : W \to W'\big) \times \prod_{w} \natural\big(\phi_{w} : \mathcal{H}_{w} \to \mathcal{H}'_{f(w)}\big)\,. \end{aligned} \tag{152}$$

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 $\mathbb{1} \;\equiv\; \mathbb{K} \,:\, \mathrm{Mod}_{\mathbb{K}}$.

Remark 2.2 (**External monoidal structures).** Given any monoidal category $(C, \otimes, \mathbb{1})$, its free coproduct completion Fam_C (of indexed sets of C-objects) inherits a corresponding "*external*" monoidal struture given by joint fiberwise product in C over the Cartesian product of index sets (for pointers see [SS23-EoS, p. 4]).

Proposition 2.3 (**Doubly closed monoidal structure of linear bundle types).** *The category* Type (151) *of linear bundle types is "doubly"* [OP99, §3] *symmetric monoidal closed* [EK66, §III][Bor94b, §6.1]*, as shown on the right, in that:*

(i) *it is cartesian closed with respect to the* external direct sum*,*

with unit object $* \;\equiv\; \left[\begin{array}{c} 0 \\ ↡ \\ * \end{array}\right] \;:\; \mathrm{Type}$

(ii) *it is non-cartesian closed symmetric monoidal with respect to the* external tensor product (cf. [RS18, Prop. 3.5])

with unit object $\mathbb{1} \;\equiv\; \left[\begin{array}{c} \mathbb{1} \\ ↡ \\ * \end{array}\right] \;:\;$ Type.

Pair types	**Function types**
$\mathrm{Hom}(X \cdot X',\, X'') \;\simeq$	$\mathrm{Hom}(X,\, [X', X''])$
$W \times W'$ 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	$\prod_w \natural(\mathcal{H}'_w \to \mathcal{H}''_w)$ 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
$\left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \times \left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right]$ $= \left[\begin{array}{c} \mathcal{H}_\bullet \oplus \mathcal{H}'_\bullet \\ \downarrow \\ W \times W' \end{array}\right]$ external direct sum	$\left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right] \to \left[\begin{array}{c} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{array}\right] =$ $\left[\begin{array}{c} \prod_{w'} \mathcal{H}''_{f(w')} \\ \downarrow \\ (f : W' \to W'') \times \\ \prod_{w'} \natural(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}) \end{array}\right]$
$\left[\begin{array}{c} \mathcal{H}_\bullet \\ \downarrow \\ W \end{array}\right] \otimes \left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right]$ $= \left[\begin{array}{c} \mathcal{H}_\bullet \otimes \mathcal{H}'_\bullet \\ \downarrow \\ W \times W' \end{array}\right]$ external tensor product	$\left[\begin{array}{c} \mathcal{H}'_\bullet \\ \downarrow \\ W' \end{array}\right] \multimap \left[\begin{array}{c} \mathcal{H}''_\bullet \\ \downarrow \\ W'' \end{array}\right] =$ $\left[\begin{array}{c} \prod_{w'} (\mathcal{H}_{w'} \multimap \mathcal{H}''_{f(w')}) \\ \downarrow \\ (f : W' \to W'') \end{array}\right]$

Remark 2.4 (**Notation for internal homs).**
(i) The arrow-notation for the hom-sets in QuType and QuType_W is that inherited from Type under the embeddings ClaType, QuType $\hookrightarrow$ Type (155), in that:

$$\natural\left(\left[\begin{array}{c} \mathcal{H} \\ ↡ \\ * \end{array}\right] \to \left[\begin{array}{c} \mathcal{H}' \\ ↡ \\ * \end{array}\right] \right) \;=\; \natural(\mathcal{H} \to \mathcal{H}')$$

$$\left(\left[\begin{array}{c} \mathcal{H} \\ ↡ \\ * \end{array}\right] \multimap \left[\begin{array}{c} \mathcal{H}' \\ ↡ \\ * \end{array}\right] \right) \;=\; (\mathcal{H} \multimap \mathcal{H}')$$

where on the right the embeddings (155) are understood.
(ii) This way, e.g. the natural hom-isomorphism expressing the closed monoidal structure on QuType reads

$$\natural(\mathcal{H} \otimes \mathcal{H}' \to \mathcal{H}'') \;\simeq\; \natural(\mathcal{H} \to (\mathcal{H}' \multimap \mathcal{H}'')) \tag{153}$$

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

$$\boxed{(W \to \mathcal{H}) \;\equiv\; \left[\begin{array}{c} 0 \\ ↡ \\ W \end{array}\right] \to \left[\begin{array}{c} \mathcal{H} \\ ↡ \\ * \end{array}\right] \;=\; \left[\begin{array}{c} \prod_W \mathcal{H} \\ ↡ \\ * \end{array}\right] \;=\; \left[\begin{array}{c} \mathbb{1}_\bullet \\ ↡ \\ W \end{array}\right] \multimap \left[\begin{array}{c} \mathcal{H} \\ ↡ \\ * \end{array}\right] \;=\; (\mathbb{1} \times W \multimap \mathcal{H})} \tag{154}$$

Proof of Prop. 2.3. By standard arguments [Schau01] we may assume the unitors and associators to be identities. The symmetric braiding is given by the evident exchange of variables:

$$\begin{aligned}
\mathrm{braid}^{\otimes}_{\mathcal{H}_W,\mathcal{H}'_{W'}} &: \begin{bmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{bmatrix} \otimes \begin{bmatrix}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{bmatrix} \longrightarrow \begin{bmatrix}\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{bmatrix} \otimes \begin{bmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{bmatrix}\\
\mathrm{braid}^{\otimes}_{\mathcal{H}_W,\mathcal{H}'_{W'}} &\equiv |\psi_w\rangle \otimes |\psi'_{w'}\rangle \;\mapsto\; |\psi'_{w'}\rangle \otimes |\psi_w\rangle
\end{aligned}$$

To see the internal-hom adjunction it is clearly sufficient (since our classical base category is ClaType $\equiv$ Set) to check the defining hom-isomorphism for the case that $W = *$ (the singleton generator of Set). In this case, we have the following sequences of natural isomorphisms:

$$\begin{aligned}
\mathrm{Hom}\left(\begin{bmatrix}\mathcal{H}\oplus\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{bmatrix}, \begin{bmatrix}\mathcal{H}''_\bullet\\ \downarrow\\ W''\end{bmatrix}\right) &\simeq (f : W' \to W'') \times \prod_{w'} \natural(\mathcal{H}\oplus\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}) && \text{by (152)}\\
&\simeq (f : W' \to W'') \times \prod_{w'} \Big(\natural(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}) \times \natural(\mathcal{H} \to \mathcal{H}''_{f(w')})\Big) && \text{by coproduct property of } \oplus\\
&\simeq (f : W' \to W'') \times \prod_{w'} \natural(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}) \times \prod_{w'} \natural(\mathcal{H} \to \mathcal{H}''_{f(w')}) && \text{since } \textstyle\prod_W(-) \text{ is right adjoint}\\
&\simeq (f : W' \to W'') \times \prod_{w'} \natural(\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}) \times \natural\Big(\mathcal{H} \to \prod_{w'} \mathcal{H}''_{f(w')}\Big) && \text{since } \mathcal{H} \to (\text{-}) \text{ is right adjoint}\\
&\simeq \mathrm{Hom}\left(\begin{bmatrix}\mathcal{H}\\ \downarrow\\ *\end{bmatrix}, \begin{bmatrix}\prod_{w'} \mathcal{H}''_{f(w')}\\ \downarrow\\ (f : W' \to W'') \times \prod\limits_{w'} (\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')})\end{bmatrix}\right) && \text{by (152)}
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{Hom}\left(\begin{bmatrix}\mathcal{H}\otimes\mathcal{H}'_\bullet\\ \downarrow\\ W'\end{bmatrix}, \begin{bmatrix}\mathcal{H}''_\bullet\\ \downarrow\\ W''\end{bmatrix}\right) &\simeq (f : W' \to W'') \times \prod_{w'} \natural\big(\mathcal{H}\otimes\mathcal{H}'_{w'} \to \mathcal{H}''_{f(w')}\big) && \text{by (152)}\\
&\simeq (f : W' \to W'') \times \prod_{w'} \natural\big(\mathcal{H} \to (\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')})\big) && \text{by (153)}\\
&\simeq (f : W' \to W'') \times \natural\big(\mathcal{H} \to \prod_{w'} (\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')})\big) && \text{since } \mathcal{H} \to (\text{-}) \text{ is right adjoint}\\
&\simeq \mathrm{Hom}\left(\begin{bmatrix}\mathcal{H}\\ \downarrow\\ *\end{bmatrix}, \begin{bmatrix}\prod_{w'} (\mathcal{H}'_{w'} \multimap \mathcal{H}''_{f(w')})\\ \downarrow\\ (f : W' \to W'')\end{bmatrix}\right) && \text{by (152)}
\end{aligned}$$

which proves the claim. □

Remark 2.5 (Linear bundle types as modules in bundles of abelian groups). Analogous formulas as in Prop. 2.3 of course hold over any commutative base ring. In particular they hold over the integers, in which case these bundle types are set-indexed families of abelian groups. From here all other cases are obtained by passage to categories of internal modules (143): Regarding the ground field $\mathbb{K}$ as a monoid internal (139) to set-indexed abelian groups

$$\begin{bmatrix}\mathbb{K}\\ \downarrow\\ \mathrm{pt}\end{bmatrix} \in \mathrm{Mon}\Big(\int_{W:\mathrm{Set}} \mathrm{Mod}^{W}_{\mathbb{Z}}\Big),$$

the $\mathbb{K}$-linear bundle types (151) are equivalently the corresponding internal modules (143):

$$\int_{W:\mathrm{Set}} \mathrm{Mod}^{W}_{\mathbb{C}} \;\simeq\; \mathrm{Mod}_{\left[\begin{smallmatrix}\mathbb{C}\\ \downarrow\\ \mathrm{pt}\end{smallmatrix}\right]}\Big(\int_{W:\mathrm{Set}} \mathrm{Mod}^{W}_{\mathbb{Z}}\Big),$$

because for a bundle over some W : Set, its external tensor product with the complex numbers, constituting the domain of the action map (142), equals the usual tensor product of bundles over W with the trivial line bundle

$$\begin{tikzcd}
\left[\begin{array}{c}\mathbb{C}\\ \twoheaddownarrow\\ \mathrm{pt}\end{array}\right] \otimes \left[\begin{array}{c}\mathcal{A}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \simeq \left[\begin{array}{c}\mathbb{C}\otimes_{\mathbb{Z}}\mathcal{A}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \arrow[r, "\rho"] & \left[\begin{array}{c}\mathcal{A}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right],
\end{tikzcd}$$

whence the action map ρ makes the indexed set of abelian groups $\mathcal{A}_\bullet$ fiberwise into a complex vector space.

While somewhat tautologous in the present case, this perspective on linear bundle types as internal modules with respect to an external tensor product becomes rather useful when we generalize in [SS23-QR], see Prop. **??**, to "Real complex module bundles" (Atiyah's Real bundles) in order to encode not just the linear structure of quantum types but also their Hermitian inner product.

Remark 2.6 (Dependent linear types). For W : ClaType we have a full embedding of the W-parameterized quantum types into the slice of all bundle types over the classical type W:

$$\begin{tikzcd}
\mathrm{QuType}_W \arrow[rr, hook] & & \mathrm{Type}_{/W} & \\
\left[\begin{array}{c}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \arrow[rr, mapsto] & & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \arrow[dr, shift left] \arrow[dr, shift right] & \\
& & & \left[\begin{array}{c}0\\ \twoheaddownarrow\\ W\end{array}\right]
\end{tikzcd}$$

exhibiting the full subcategory of the slice on those objects whose fibers are purely quantum:

$$\begin{tikzcd}
\left[\begin{array}{c}\mathcal{H}_w\\ \twoheaddownarrow\\ \{w\}\end{array}\right] \arrow[rr, shift left] \arrow[rr, hook, shift right] \arrow[dr, shift left] \arrow[dr, shift right] & & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \arrow[dr, shift left] \arrow[dr, shift right] & \\
& \left[\begin{array}{c}0\\ \twoheaddownarrow\\ \{w\}\end{array}\right] \arrow[rr, shift left] \arrow[rr, hook, shift right] & & \left[\begin{array}{c}0\\ \twoheaddownarrow\\ W\end{array}\right].
\end{tikzcd}$$

Classical and Quantum Modality.

Proposition 2.7 (Reflective subcategories of purely classical/quantum modal types). *The category of Def. 2.1 has monadic* (98) *reflective subcategory inclusions as follows:*

$$\begin{tikzcd}
W & \longmapsfrom & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] & & \bigoplus\limits_{w}\mathcal{H}_w & \longmapsfrom & \left[\begin{array}{c}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] \\
\mathrm{ClaType} \arrow[rr, hook, shift right] & \bot & \mathrm{Type} \arrow[ll, shift right] \arrow[loop right, "\natural\ \text{classically}"] & & \mathrm{QuType} \arrow[rr, hook, shift right] & \bot & \mathrm{Type} \arrow[ll, shift right] \arrow[loop right, "\rhd\ \text{quantumly}"] \\
W & \longmapsto & \left[\begin{array}{c}0_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] & & \mathcal{H} & \longmapsto & \left[\begin{array}{c}\mathcal{H}\\ \twoheaddownarrow\\ *\end{array}\right]
\end{tikzcd} \tag{155}$$

Moreover, the induced classical/quantum-modalities are strong monads (77) with respect to the monoidal structures of Prop. 2.3, whence we have `return`- and `bind`-operations (67) as follows, using (162):

$$
\begin{array}{c||llll}
\text{Classically} \; \natural & \texttt{return}^\natural_W \; : \left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \to \left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] & & \texttt{bind}^\natural \; : \left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \to \left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]\right) \to \left(\left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \to \left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]\right) & \\
 & \texttt{return}^\natural_W \equiv |\psi_w\rangle \; \mapsto \; 0_w & & \texttt{bind}^\natural \equiv \; (|\psi_w\rangle \mapsto 0_{f(w)}, 0) \; \mapsto \; (0_w \; \mapsto \; 0_{f(w)}, 0) & \\
\hline
\text{Quantumly} \; \triangleright & \texttt{return}^\triangleright \; \mathrel{⸰} \left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \multimap \left[\begin{matrix}\oplus_w \mathcal{H}_w\\ \twoheaddownarrow \\ *\end{matrix}\right] & & \texttt{bind}^\triangleright \; \mathrel{⸰} \left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \multimap \left[\begin{matrix}\mathcal{H}'\\ \twoheaddownarrow \\ *\end{matrix}\right]\right) \multimap \left(\left[\begin{matrix}\oplus_w \mathcal{H}_w\\ \twoheaddownarrow \\ *\end{matrix}\right] \multimap \left[\begin{matrix}\mathcal{H}'\\ \twoheaddownarrow \\ *\end{matrix}\right]\right) & \\
 & \texttt{return}^\triangleright \equiv |\psi_w\rangle \; \mapsto \; \underset{w'}{\oplus} \delta_w^{w'} |\psi_w\rangle & & \texttt{bind}^\triangleright \equiv \; (|\psi_w\rangle \mapsto F_w |\psi_w\rangle) \; \mapsto \; \Big(\underset{w}{\oplus} |\psi_w\rangle \mapsto \sum_w F_w |\psi_w\rangle \Big) &
\end{array}
\tag{156}
$$

Proof. It is evident that the inclusions are fully faithful and reflective. Formally we may check the required hom-isomorphisms (75) using (152):

$$
\mathrm{Hom}\left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right], \left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]\right) \simeq \mathrm{Hom}(W, W') \times \prod_w \mathrm{Hom}(\mathcal{H}_w, 0) \simeq \mathrm{Hom}(W,\, W') \simeq \mathrm{Hom}\left(\left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right], \left[\begin{matrix}0_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]\right)
$$

$$
\mathrm{Hom}\left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right], \left[\begin{matrix}\mathcal{H}'\\ \twoheaddownarrow \\ *\end{matrix}\right]\right) \simeq \mathrm{Hom}(\mathcal{H},\, *) \times \prod_w \mathrm{Hom}(\mathcal{H}_w,\, \mathcal{H}') \simeq \mathrm{Hom}\Big(\coprod_w \mathcal{H}_w,\, \mathcal{H}'\Big) \simeq \mathrm{Hom}\left(\left[\begin{matrix}\oplus_w \mathcal{H}_w\\ \twoheaddownarrow \\ *\end{matrix}\right], \left[\begin{matrix}\mathcal{H}'\\ \twoheaddownarrow \\ *\end{matrix}\right]\right)
$$

Monadicity follows because every reflective inclusion is monadic (e.g. [Bor94b, Cor. 4.2.4]). Alternatively, we may invoke the monadicity theorem in the form (99): 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 [SS23-EoS, Prop. A.9].

Finally, to check that the induced monads are strong, we may equivalently check that they are monoidal (78): 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 are monoidal, hence that these squares commute, which is immediate in components (156):

$$
\begin{tikzcd}
\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \times \left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \arrow[r, "\mathrm{ret}^\natural"] \arrow[ddd, equal] & \natural\left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \times \left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right]\right) \arrow[d, phantom, "\wr \quad \text{by Prop. 2.3}"] \\
 & \natural\left[\begin{matrix}\mathcal{H}_\bullet \oplus \mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W \times W'\end{matrix}\right] \arrow[d, phantom, "\wr \quad \text{by (155)}"] \\
 & W \times W' \arrow[d, phantom, "\wr \quad \text{by (155)}"] \\
\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \times \left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \arrow[r, "\mathrm{ret}^\natural \times \mathrm{ret}^\natural"] & \natural\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \times \natural\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right]
\end{tikzcd}
\qquad
\begin{tikzcd}
\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \otimes \left[\begin{matrix}\mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right] \arrow[r, "\mathrm{ret}^\triangleright"] \arrow[ddddd, equal] & \triangleright\left(\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \otimes \left[\begin{matrix}\mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]\right) \arrow[d, phantom, "\equiv \quad \text{from Prop. 2.3}"] \\
 & \triangleright\left[\begin{matrix}\mathcal{H}_\bullet \otimes \mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W \times W'\end{matrix}\right] \arrow[d, phantom, "\wr \quad \text{by (155)}"] \\
 & \bigoplus_{(w,w')} (\mathcal{H}_w \otimes \mathcal{H}'_{w'}) \arrow[d, phantom, "\wr"] \\
 & \bigoplus_w \bigoplus_{w'} (\mathcal{H}_w \otimes \mathcal{H}'_{w'}) \arrow[d, phantom, "\wr \quad \text{distributivity}"] \\
 & \Big(\bigoplus_w \mathcal{H}_w\Big) \otimes \Big(\bigoplus_{w'} \mathcal{H}'_{w'}\Big) \arrow[d, phantom, "\wr \quad \text{by (155)}"] \\
\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \otimes \left[\begin{matrix}\mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right] \arrow[r, "\mathrm{ret}^\triangleright \otimes \mathrm{ret}^\triangleright"] & \triangleright\left[\begin{matrix}\mathcal{H}_\bullet\\ \twoheaddownarrow \\ W\end{matrix}\right] \otimes \triangleright\left[\begin{matrix}\mathcal{H}'_\bullet\\ \twoheaddownarrow \\ W'\end{matrix}\right]
\end{tikzcd}
$$

□

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

In fact, the purely classical types are also coreflective, whence the classical-modality $\natural$ is in fact a *bireflective Frobenius modality* [FHPTST99, Def. 8]:

Proposition 2.8 (**Coreflection of classical types among linear bundle types).** *We have an ambidextrous reflection:*

$$\begin{tikzcd} \mathrm{ClaType} \arrow[r, hook] & \mathrm{Type} \arrow[l, shift right=2.5, "\natural"'] \arrow[l, shift left=2.5, "\natural"] \end{tikzcd} \;. \tag{157}$$

Quantization and Exponential modality. Composing the Cartesian hom-adjunction for $\mathbb{1}$ (from Prop. 2.3) with the classicality-coreflection (157) gives another adjunction between linear bundle types and purely classical types:

$$\begin{tikzcd}[column sep=large] W \arrow[rr, mapsto] & & \begin{bmatrix}\mathbb{1}_\bullet \\ \downarrow \\ W\end{bmatrix} \\ \mathrm{ClaType} \arrow[r, hook, shift left=1.5] \arrow[r, phantom, "\perp"] \arrow[rr, bend left=40] & \mathrm{Type} \arrow[l, shift left=1.5, "\natural"] \arrow[r, shift left=1.5, "\mathbb{1}\times(-)"] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift left=1.5, "\mathbb{1}\to(-)"] \arrow[ll, bend left=40] \\ (w:W)\times\natural(\mathbb{1}\to\mathcal{H}_w) & & \begin{bmatrix}\mathcal{H}_\bullet \\ \downarrow \\ W\end{bmatrix} \arrow[ll, mapsto'] \end{tikzcd} \tag{158}$$

(cf. Rem. 2.11). Further composing (158) with the reflection of purely quantum types (155) gives (cf. Rem. 2.10):

Proposition 2.9 (**Quantization and Classicization).**
(i) *We have a pair of adjoint functors between purely classical and purely quantum types* (155) *of this form*

$$\begin{tikzcd}[column sep=huge] W \arrow[rr, mapsto] & & \underset{W}{\bigoplus}\mathbb{1} \\ \mathrm{ClaType} \arrow[r, shift left=1.5, "\mathbb{1}\times(-)"] \arrow[r, phantom, "\perp"] \arrow[rr, bend left=40, "\overset{\text{quantized}}{\mathrm{Q}} \;\equiv\; \overset{\text{motive}}{\Sigma^\infty_+}"] & \mathrm{Type} \arrow[l, shift left=1.5, "\natural(\mathbb{1}\to(-))"] \arrow[r, shift left=1.5, "\triangleright"] \arrow[r, phantom, "\perp"] & \mathrm{QuType} \arrow[l, hook', shift left=1.5] \arrow[ll, bend left=40, "\underset{\text{classicized}}{\mathrm{C}} \;\equiv\; \Omega^\infty_+"] \arrow[loop right, "!\;\;\text{exponential modality}"] \\ \natural(\mathbb{1}\to\mathcal{H}) & & \mathcal{H} \arrow[ll, mapsto'] \end{tikzcd} \tag{159}$$

where the composite $! \equiv QC$ *is the "exponential modality"* (Rem. 2.10).
(ii) *These are monoidal with respect to the classical/quantum monoidal structures (Prop. 2.3) via natural transformations of the following form:*

$$\begin{array}{rcl} W, W' : \mathrm{ClaType} & \vdash & (\mathrm{Q}W)\otimes(\mathrm{Q}W') \;\simeq\; \mathrm{Q}(W\times W') \\ \mathcal{H},\mathcal{H}' : \mathrm{QuType} & \vdash & (\mathrm{C}\mathcal{H})\times(\mathrm{C}\mathcal{H}') \;\simeq\; \mathrm{C}(\mathcal{H}\times\mathcal{H}') \\ \mathcal{H},\mathcal{H}' : \mathrm{QuType} & \vdash & (\mathrm{C}\mathcal{H})\times(\mathrm{C}\mathcal{H}') \;\to\; \mathrm{C}(\mathcal{H}\otimes\mathcal{H}') \end{array} \tag{160}$$

$$\mathrm{Q}* \;\simeq\; \mathbb{1}\,, \qquad \mathrm{C}0 \;\simeq\; \mathbb{1}\,, \qquad \mathrm{C}\mathbb{1} \;\to\; \mathbb{1}\,. \tag{161}$$

(iii) *In particular, the induced modality* (159) *sends (direct) sums to (tensor) products*

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

and zero (objects) to unit (objects)

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

as befits an exponential map.

Proof. The adjunction itself is the composite of (158) with (155), 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 $\oplus_W$:

$$(\mathrm{Q}W)\otimes(\mathrm{Q}W') \;\equiv\; (\oplus_W\mathbb{1})\otimes(\oplus_{W'}\mathbb{1}) \;\equiv\; \underset{W\times W'}{\oplus}(\mathbb{1}\otimes\mathbb{1}) \;=\; \underset{W\times W'}{\oplus}\mathbb{1} \;\equiv\; \mathrm{Q}(W\times W')\,.$$

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:

$$\begin{aligned}
(\mathrm{C}\mathcal{H})\times(\mathrm{C}\mathcal{H}) &\equiv \natural(\mathbb{1}\to\mathcal{H})\times\natural(\mathbb{1}\to\mathcal{H}') \\
&\simeq \natural\big((\mathbb{1}\to\mathcal{H})\times(\mathbb{1}\to\mathcal{H}')\big) && \text{since } \natural \text{ is right adjoint} \\
&\simeq \natural\big((\mathbb{1}\to(\mathcal{H}\times\mathcal{H}'))\big) && \text{since } \mathbb{1}\to(\text{-}) \text{ is right adjoint} \\
&\equiv \mathrm{C}(\mathcal{H}\times\mathcal{H}') \\
&\to \mathrm{C}(\mathcal{H}\otimes\mathcal{H}') && \text{univ. bilin.}
\end{aligned}$$

□

Remark 2.10 (Exponential modality, traditionally). Prop 2.9 recovers – via dependent linear type formations – the *exponential modality* (25) usually postulated in linear logic/type theory (Lit. 1.4). In the model QuType ≡ $\mathrm{Mod}_{\mathbb{K}}$ (151), the operation $\mathcal{H}\mapsto\natural(\mathbb{1}\to\mathcal{H})$ (158) produces the *underlying set* of vectors in the vector space $\mathcal{H}$, whence the exponential modality (159) sends a vector space to the linear span of its underlying set of vectors

$$\mathcal{H} : \mathrm{Mod}_{\mathbb{K}} \qquad \vdash \qquad !\mathcal{H} = \underset{\mathcal{H}}{\oplus}\mathbb{1}\,.$$

As an aside it is interesting that in the homotopy-theoretic semantics of `HoTT` in parameterized spectra, the exponential modality (159) on, in that case, QuType ≡ Spectra is known to behave like an exponential function in the sense of "Goodwillie calculus", see [ACh19, Ex. 2.6].

Remark 2.11 (Exponential modality, in `Quipper`). In contrast to Rem. 2.10, beware that the literature on `Quipper` (Lit. 1.5) instead chooses to write "!" for the comonad induced in (158), see [RS18, §3.5, §3.7 & Def. 3.7][FKS20, p. 9].

Remark 2.12 (Role of the exponential modality). Below in §3 we will not have much use for the exponential modality: Its purpose in traditional linear logic/type theory is to get access to a stand-in for classical types in a theory that natively only knows about linear types. But this becomes a moot point in a classically-dependent linear type theory like `LHoTT`, as formally reflected by the above construction showing that the exponential modality is derivable from dependent linear type formation. For our purpose here this construction serves to show that `LHoTT` is backwards-compatible with previous discussion of linear type theory via an exponential modality, cf. [Ri23, p. 9].

Quantum type declaration. For transparent distinction between the classical and quantum monoidal structures from Prop. 2.3 it is convenient to use, besides the standard notation for

- the classical type declaration in the empty context

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

  which is equivalently type declaration in the context of the cartesian monoidal unit $*$: ClaType

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

also notation for

- a linear (quantum) type declaration

  $$\vdash \qquad |\psi\rangle \mathrel{\substack{\circ\\\circ}} \mathcal{H}\,,$$

  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 $W \to \mathcal{H}$ (154).

Data	Declaration	Semantics
Classical	$\vdash\; W \;:\; \mathrm{ClaType}$ $\vdash\; w \;:\; W$	$\left[\begin{array}{c}0\\ \downarrow\\ *\end{array}\right] \begin{array}{c}\xrightarrow{\;0_w\;}\\ \\ \xrightarrow{\;w\;}\end{array} \left[\begin{array}{c}0_\bullet\\ \downarrow\\ W\end{array}\right]$
Quantum	$\vdash\; \mathcal{H} \;:\; \mathrm{QuType}$ $\vdash\; \vert\psi\rangle \;\mathrel{\overset{\circ}{\circ}}\; \mathcal{H}$ $\equiv$ $\vdash\; \vert\psi\rangle \;:\; \mathbb{1} \to \mathcal{H}$	$\left[\begin{array}{c}\mathbb{1}\\ \downarrow\\ *\end{array}\right] \begin{array}{c}\xrightarrow{\;\vert\psi\rangle\;}\\ \\ \xrightarrow{\;*\;}\end{array} \left[\begin{array}{c}\mathcal{H}\\ \downarrow\\ *\end{array}\right]$
Quantized	$\vdash\; W \;:\; \mathrm{ClaType}$ $\vdash\; \mathcal{H} \;:\; \mathrm{QuType}$ $\vdash\; \sum_w \vert w\rangle \;\mathrel{\overset{\circ}{\circ}}\; W \to \mathcal{H}$ $\equiv$ $\vdash\; \sum_w \vert w\rangle \;:\; \mathbb{1} \to (W \to \mathcal{H})$	$\left[\begin{array}{c}\mathbb{1}\\ \twoheaddownarrow\\ *\end{array}\right] \begin{array}{c}\xrightarrow{\;\sum_w \vert w\rangle\;}\\ \\ \xrightarrow{\;*\;}\end{array} \left[\begin{array}{c}\prod_W \mathcal{H}\\ \twoheaddownarrow\\ *\end{array}\right]$

(162)

We will have much use in §3 for the following:

Definition 2.13 (Quantization modality). We will regard quantization (159) as the *relative* monad (101) obtained by restricting (102) the quantum-modality $\triangleright$ (2.7) along precomposition with (158):

$$\begin{array}{ccccccc} \mathrm{Q} \;: & \mathrm{ClaType} & \xrightarrow{\;\mathbb{1}\times(\text{-})\;} & \mathrm{Type} & \xrightarrow{\;\triangleright\;} & \mathrm{Type} \\ & W & \mapsto & \left[\begin{array}{c}\mathbb{1}_\bullet\\ \twoheaddownarrow\\ W\end{array}\right] & \mapsto & \underset{W}{\oplus}\mathbb{1} \end{array} \tag{163}$$

This (just) means that we take the `return`- and `bind`-operations (67) of Q to be special instances of those of $\triangleright$, as follows, where we use the linear type declaration from (162):

$$\begin{array}{llll} \mathtt{return}^{\mathrm{Q}}_{W} \;\mathrel{\overset{\circ}{\circ}}\; (\mathbb{1}\times W \multimap \mathrm{Q}W) & & \mathtt{bind}^{\mathrm{Q}}_{W,\mathrm{Q}W'} \;\mathrel{\overset{\circ}{\circ}}\; (\mathbb{1}\times W \multimap \mathrm{Q}W') \multimap (\mathrm{Q}W \multimap \mathrm{Q}W') \\ \mathtt{return}^{\mathrm{Q}}_{W} \;\equiv\; \mathtt{return}^{\triangleright}_{\mathbb{1}\times W} & & \mathtt{bind}^{\mathrm{Q}}_{W,\mathrm{Q}W'} \;\equiv\; \mathtt{bind}^{\triangleright}_{\mathbb{1}\times W,\mathrm{Q}W'} \end{array} \tag{164}$$

But in these special cases of $\triangleright$-operations we may, by (154), equivalently write this pleasantly suggestively as follows:

$$\boxed{\begin{array}{c|ll} \textbf{Quantized}\ \mathrm{Q} & \mathtt{return}^{\mathrm{Q}}_{W} \;\mathrel{\overset{\circ}{\circ}}\; (W \to \mathrm{Q}W) & \mathtt{bind}^{\mathrm{Q}} \;\mathrel{\overset{\circ}{\circ}}\; (W \to \mathrm{Q}W') \multimap (\mathrm{Q}W \longrightarrow\!\!\!\circ\; \mathrm{Q}W') \\ & \mathtt{return}^{\mathrm{Q}}_{W} \;\equiv\; w \mapsto \vert w\rangle & \mathtt{bind}^{\mathrm{Q}} \;\equiv\; (w \mapsto \vert\psi_w\rangle) \mapsto \Big(\sum_w q_w \vert w\rangle \;\mapsto\; \sum_w q_w \vert\psi_w\rangle\Big) \end{array}} \tag{165}$$

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 2.14 (Type of qbits). The notation for the quantization-monad (Def. 2.13) is such as to reproduce the standard notation "QBit" for the type of q-bits (e.g. [NC00, §1.2], often also "qubit", e.g. [Ri21]) as the quantum analog of the type $\mathrm{Bit} \equiv \{0,1\}$ of classical bits (cf. [TQP, (110)]):

$$\mathrm{QBit} \;\equiv\; \mathrm{Q}(\mathrm{Bit}) \;\equiv\; \triangleright(\mathbb{1}_{\mathrm{Bit}}) \;\equiv\; \underset{\mathrm{Bit}}{\bigoplus}\, \underset{\mathrm{Bit}}{\mathbb{1}} \;\equiv\; \underset{\{0,1\}}{\bigoplus}\, \underset{\{0,1\}}{\mathbb{1}} \;\equiv\; \mathbb{1}_0 \oplus \mathbb{1}_1 \;=\; \{q_0\,|0\rangle + q_1\,|1\rangle\}. \tag{166}$$

Similarly we have the restriction of the quantum-modality to tensor products, hence to entangled states:

Definition 2.15 (Entanglement modality). Recalling the cartesian product of classical types and the tensor product (Prop. 2.3) of quantized linear types (Def. 2.13)

$$\begin{array}{rl} (\text{-})\times(\text{-}) \;: & \mathrm{ClaType}\times\mathrm{ClaType} \longrightarrow \mathrm{Type} \\ \mathrm{Q}(\text{-})\otimes\mathrm{Q}(\text{-}) \;: & \mathrm{ClaType}\times\mathrm{ClaType} \longrightarrow \mathrm{Type} \end{array}$$

the restriction of the $\triangleright$-monad along $\mathrm{Q}(-)\otimes\mathrm{Q}(-)$ yields a relative monad of this form (recalling that $\triangleright$ is the identity on linear types)

$$\boxed{\begin{array}{c|lll} & \mathtt{return}^{\otimes}_{(B_1,B_2)} & \mathrel{\overset{\circ}{\circ}} & B_1\times B_2 \to \mathrm{Q}B_1\otimes\mathrm{Q}B_2 \\ & \mathtt{return}^{\otimes}_{(B_1,B_2)} & \equiv & (b_1,b_2)\mapsto |b_1\rangle\otimes|b_2\rangle \\ \textbf{entangled} & & & \\ & \mathtt{bind}^{\otimes}_{(B_1,B_2),\mathcal{H}} & \mathrel{\overset{\circ}{\circ}} & (B_1\times B_2 \to \mathcal{H}) \multimap (\mathrm{Q}B_1\otimes\mathrm{Q}B_2 \longrightarrow\!\!\!\circ\; \mathcal{H}) \\ & \mathtt{bind}^{\otimes}_{(B_1,B_2),\mathcal{H}} & \equiv & ((b_1,b_2)\mapsto |\psi_{b_1,b_2}\rangle) \mapsto \Big(\big(\sum_{b_1,b_2} q_{b_1,b_2}\cdot|b_1\rangle\otimes|b_2\rangle\big) \mapsto \sum_{b_1,b_2} q_{b_1,b_2}\cdot|\psi_{b_1,b_2}\rangle\Big) \end{array}} \tag{167}$$

In summary so far, we have the following fundamental quantum modalities:

The Quantum/Classical Divide		
Modality	Idempotent **monad**	**Pure effect**
Classical	$\natural \;:\; \mathrm{Type} \twoheadrightarrow \mathrm{ClaType} \hookrightarrow \mathrm{Type}$ $\natural \equiv \left[\begin{smallmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{smallmatrix}\right] \mapsto W \mapsto \left[\begin{smallmatrix}0_\bullet\\ \downarrow\\ W\end{smallmatrix}\right]$ (strong wrt ×)	$\mathrm{ret}^{\natural}_{\mathcal{H}_\bullet} \;:\; \mathcal{H}_\bullet \longrightarrow \natural\mathcal{H}_\bullet$ $\left[\begin{smallmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{smallmatrix}\right] \begin{array}{c}\xrightarrow{0}\\ \xrightarrow{\mathrm{id}}\end{array} \left[\begin{smallmatrix}0_\bullet\\ \downarrow\\ W\end{smallmatrix}\right]$
Quantum	$\triangleright \;:\; \mathrm{Type} \twoheadrightarrow \mathrm{QuType} \hookrightarrow \mathrm{Type}$ $\triangleright \equiv \left[\begin{smallmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{smallmatrix}\right] \mapsto \underset{W}{\bigoplus}\mathcal{H}_\bullet \mapsto \left[\begin{smallmatrix}\underset{W}{\bigoplus}\mathcal{H}_\bullet\\ \downarrow\\ *\end{smallmatrix}\right]$ (strong wrt ⊗)	$\mathrm{ret}^{\triangleright}_{\mathcal{H}_\bullet} \mathrel{\substack{\circ\\ \circ}} \mathcal{H}_\bullet \multimap \triangleright\mathcal{H}_\bullet$ $\left[\begin{smallmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{smallmatrix}\right] \begin{array}{c}\xrightarrow{\mathrm{ret}^{\lozenge_W}_{\mathcal{H}_\bullet}}\\ \xrightarrow{p_W}\end{array} \left[\begin{smallmatrix}\underset{W}{\bigoplus}\mathcal{H}_\bullet\\ \downarrow\\ *\end{smallmatrix}\right]$
Quantized	$\mathrm{Q} \;:\; \mathrm{ClaType} \longrightarrow \mathrm{QuType} \hookrightarrow \mathrm{Type}$ $\mathrm{Q} \;\equiv\; W \;\mapsto\; \triangleright(\mathbb{1}_W)$ (relative monad)	$\mathrm{ret}^{\mathrm{Q}}_{\mathcal{H}_\bullet} \mathrel{\substack{\circ\\ \circ}} W \longrightarrow \mathrm{Q}W$ $\left[\begin{smallmatrix}\mathbb{1}_\bullet\\ \downarrow\\ W\end{smallmatrix}\right] \begin{array}{c}\xrightarrow{\mathrm{ret}^{\lozenge_W}_{\mathbb{1}_\bullet}}\\ \xrightarrow{p_W}\end{array} \left[\begin{smallmatrix}\underset{W}{\bigoplus}\mathbb{1}\\ \twoheaddownarrow\\ *\end{smallmatrix}\right]$
Entangled	$\mathrm{Q}(\text{-}) \otimes \mathrm{Q}(\text{-}) \;:\; \mathrm{ClaType} \times \mathrm{ClaType} \longrightarrow \mathrm{QuType}$ (relative monad)	$\mathrm{ret}^{\otimes} \mathrel{\substack{\circ\\ \circ}} (W_1, W_2) \longrightarrow \mathrm{Q}W_1 \otimes \mathrm{Q}W_2$ $\left[\begin{smallmatrix}\mathbb{1}_\bullet\\ \downarrow\\ W_1 \times W_2\end{smallmatrix}\right] \begin{array}{c}\xrightarrow{\mathrm{ret}^{\lozenge_{W_1\times W_2}}_{\mathcal{E}}}\\ \xrightarrow{p_{W_1\times W_2}}\end{array} \left[\begin{smallmatrix}\mathrm{Q}W_1 \otimes \mathrm{Q}W_2\\ \twoheaddownarrow\\ *\end{smallmatrix}\right]$

Base change and dependent classical/linear type formation. In parameterized generalization of the reflection of quantum types inside all bundle types (Prop. 2.7), also the W-parameterized linear types (151) are reflective in the *slice category* $\mathrm{Type}_{/W}$ of bundle types over the given classical type $W = \left[\begin{smallmatrix}0_\bullet\\ \twoheaddownarrow\\ W\end{smallmatrix}\right]$:

$$\begin{tikzcd}
\left[\begin{smallmatrix}\underset{p'(w')=w}{\bigoplus}\mathcal{H}'_{w'}\\ \twoheaddownarrow\\ (w : W)\end{smallmatrix}\right] & \begin{array}{c}[\mathcal{H}'_\bullet \to W']\\ \downarrow \qquad \downarrow{\scriptstyle p'}\\ [0_\bullet \to W]\end{array} \arrow[l, mapsto] \\
\mathrm{QuType}_W \arrow[r, hook, shift right] \arrow[r, phantom, "\bot"] & \mathrm{Type}_{/W} \arrow[l, shift right] \arrow[loop right, "\triangleright_W", "\textbf{W-quantumly}"'] \\
\left[\begin{smallmatrix}\mathcal{H}_\bullet\\ \twoheaddownarrow\\ W\end{smallmatrix}\right] \arrow[r, mapsto] & \begin{array}{c}[\mathcal{H}_\bullet \to W]\\ \downarrow \qquad \downarrow\\ [0_\bullet \to W]\end{array}
\end{tikzcd} \tag{168}$$

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

Proposition 2.16 (Classical base change for linear bundle types). *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. 2.1)*

$$\begin{tikzcd}
W \arrow[r, "p"] & \Gamma \\
\mathrm{Type}_{/W} & \mathrm{Type}_{/\Gamma} \arrow[l, "W\times_\Gamma(\text{-})", "\textbf{context extension}"'] \\
\begin{array}{c}[\mathcal{H}'_{w'} \to (w' : W'_{p(w)})]\\ \downarrow \qquad\qquad \downarrow\\ [0_w \to (w : W)]\end{array} & \begin{array}{c}[\mathcal{H}'_\bullet \to W']\\ \downarrow \qquad \downarrow\\ [0_\bullet \to \Gamma]\end{array} \arrow[l, mapsto]
\end{tikzcd}$$

has both a left adjoint ("dependent coproduct" [30]) and a right adjoint ("dependent product"), given as follows:

$$\begin{tikzcd}
{[\mathcal{H}'_\bullet} \arrow[r] \arrow[d] & {W']} \arrow[d, "p'"] & {} \arrow[r, phantom, "\mapsto"] & {} & {\Big[\mathcal{H}'_{w'_w}} \arrow[r] \arrow[d] & {\big((w, w'_w) : \coprod\limits_{p(w)=\gamma} W'_w\big)\Big]} \arrow[d] \\
{[0_\bullet} \arrow[r] & {W]} & & & {[0_\bullet} \arrow[r] & {(\gamma : \Gamma)]}
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=huge]
\mathrm{Type}_{/W} \arrow[r, shift left=4ex, "\textbf{dependent coproduct}", "\coprod_W"'] \arrow[r, shift right=4ex, "\prod_W", "\textbf{dependent product}"'] \arrow[r, phantom, shift left=2ex, "\perp"] \arrow[r, phantom, shift right=2ex, "\perp"] & \mathrm{Type}_{/\Gamma} \arrow[l, "W\times_\Gamma(\text{-})" description]
\end{tikzcd} \tag{169}$$

$$\begin{tikzcd}
{[\mathcal{H}'_\bullet} \arrow[r] \arrow[d] & {W']} \arrow[d] & {} \arrow[r, phantom, "\mapsto"] & {} & {\Big[\prod\limits_{p(w)=\gamma} \mathcal{H}'_{w'_w}} \arrow[r] \arrow[d] & {\Big(w'_\bullet : \prod\limits_{p(w)=\gamma} W'_w\Big)\Big]} \arrow[d] \\
{[0_\bullet} \arrow[r] & {W]} & & & {[0_\bullet} \arrow[r] & {(\gamma : \Gamma)]}
\end{tikzcd}$$

Proof. We may formally check the hom-isomorphisms, using (152). It is sufficient to consider the case that $\Gamma = *$:

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

□

The (co)restriction of the base change adjoint triple (169) along the reflective inclusion of W-quantum types (168) yields base change of dependent linear types:

$$\begin{tikzcd}[column sep=large]
\mathrm{Type}_{/W} \arrow[rr, shift left=4ex, "\coprod_W"] \arrow[rr, shift right=4ex, "\prod_W"'] \arrow[rr, phantom, shift left=2ex, "\perp"] \arrow[rr, phantom, shift right=2ex, "\perp"] & & \mathrm{Type}_{/\Gamma} \arrow[ll, "W\times_\Gamma(\text{-})" description] \arrow[dr, shift left, "\triangleright_\Gamma"] \arrow[dr, phantom, "\perp"] & \\
& \mathrm{QuType}_W \arrow[ul, hook] \arrow[rr, shift left=4ex, "\oplus_W"] \arrow[rr, shift right=4ex, "\prod_W"'] \arrow[rr, phantom, shift left=2ex, "\perp"] \arrow[rr, phantom, shift right=2ex, "\perp"] & & \mathrm{QuType}_\Gamma \arrow[ul, hook, shift left] \arrow[ll, "\underset{W}{\mathbb{1}}\otimes(\text{-})" description]
\end{tikzcd} \tag{170}$$

$$(w : W \ \vdash \ \mathcal{H}_w) \qquad \mapsto \qquad (\gamma : \Gamma \ \vdash \ \prod_{p(w)=\gamma} \mathcal{H}_w)$$

Now something special happens: Since $\mathrm{Mod}_{\mathbb{K}}$ is an additive category, it has *biproducts*, meaning that finite coproducts are finite products. This is a key aspect of what it means for its objects to be *linear* types.

Proposition 2.17 (Ambidexterity). *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 the base change adjunction* (170) *on linear types becomes ambidextrous:*

$$\Gamma : \mathrm{ClaType}, \quad W : \mathrm{ClaType}^{\mathrm{fin}} \qquad \vdash \qquad
\begin{array}{c}
(w : W \ \vdash \ \mathcal{H}_w) \qquad \longmapsto \qquad (\gamma : \Gamma \ \vdash \ \bigoplus\limits_{p(w)=\gamma} \mathcal{H}_w) \\
\begin{tikzcd}
\mathrm{QuType}_W \arrow[r, shift left=4ex, "\oplus_W"] \arrow[r, shift right=4ex, "\oplus_W"'] \arrow[r, phantom, shift left=2ex, "\perp"] \arrow[r, phantom, shift right=2ex, "\perp"] & \mathrm{QuType}_\Gamma \arrow[l, "\underset{W}{\mathbb{1}}\otimes(\text{-})" description]
\end{tikzcd}
\end{array} \tag{171}$$

---

[30] 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* $\oplus_W$ of linear types.

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 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. [SS23-EoS, p. 41]):

Definition 2.18 (**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:

$$\begin{tikzcd}
\mathrm{ClaType} \arrow[r, hook] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"'] & \mathrm{Type} \arrow[l, shift right=4, "\natural"'] \arrow[l, shift left=4, "\natural"] \arrow[loop right, "\natural"]
\end{tikzcd} \tag{172}$$

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

$$\mathrm{ClaType}^{\mathrm{fin}} \hookrightarrow \mathrm{ClaType} \tag{173}$$

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

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

$$\mathrm{QuType}_W \;\equiv\; \natural/W \;=\; \left\{ \begin{bmatrix}\mathcal{H}_\bullet\\ \downarrow\\ W\end{bmatrix} \begin{matrix}\xrightarrow{\;\phi_\bullet\;}\\ \\ =\!=\!=\!=\end{matrix} \begin{bmatrix}\mathcal{H}'_\bullet\\ \downarrow\\ W\end{bmatrix}\right\}, \tag{174}$$

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

$$\text{for} \qquad B \xrightarrow{f} B' \qquad \text{we have} \qquad \begin{tikzcd}
\mathrm{QuType}_W \arrow[r, shift left=4, "f_!"] \arrow[r, shift right=4, "f_*"'] \arrow[r, phantom, shift left=2, "\perp"] \arrow[r, phantom, shift right=2, "\perp"'] & \mathrm{QuType}_{W'} \arrow[l, "f^*"']
\end{tikzcd} \tag{175}$$

such that the following conditions hold:

(a) **Linearity**: the left and right base change along finite types $W \xrightarrow{p_W} *$ (see (173)) are naturally equivalent:

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

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

$$(f^* \circ g^*) \;\simeq\; g^* \circ f^* \quad \text{and} \quad \mathrm{id}^* = \mathrm{id} \tag{176}$$

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

$$f^*(\mathcal{H} \underset{W'}{\otimes} \mathcal{H}')_\bullet \;\simeq\; \big(f^*(\mathcal{H}) \underset{W'}{\otimes} f^*(\mathcal{H}')\big)_\bullet \tag{177}$$

(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:

$$\text{For} \;\; \begin{tikzcd}[column sep=small]
 & B \times_{B_0} B' \arrow[dl, "\mathrm{pr}_B"'] \arrow[dr, "\mathrm{pr}_{B'}"] & \\
B \arrow[dr, "p_B"'] & (\mathrm{pb}) & B' \arrow[dl, "p_{B'}"] \\
 & B_0 &
\end{tikzcd} \;\; \text{we have} \;\; \begin{tikzcd}[column sep=small]
 & \mathrm{QuType}_{B \times_{B_0} B'} \arrow[dl, "(\mathrm{pr}_B)_!"'] & \\
\mathrm{QuType}_B & & \mathrm{QuType}_{B'} \arrow[ul, "(\mathrm{pr}_{B'})^*"'] \arrow[dl, "(p_{B'})_!"] \\
 & \mathrm{QuType}_{B_0} \arrow[ul, "(p_B)^*"] &
\end{tikzcd} \;\; \text{and} \;\; \begin{tikzcd}[column sep=small]
 & \mathrm{QuType}_{B \times_{B_0} B'} \arrow[dl, "(\mathrm{pr}_B)_*"'] & \\
\mathrm{QuType}_B & & \mathrm{QuType}_{B'} \arrow[ul, "(\mathrm{pr}_{B'})^*"'] \arrow[dl, "(p_{B'})_*"] \\
 & \mathrm{QuType}_{B_0} \arrow[ul, "(p_B)^*"] &
\end{tikzcd} \tag{178}$$

(e) **Frobenius reciprocity / projection formula**: the left pushforward of a tensor with a pullback is naturally equivalent to the tensor with the left pushforward (equivalent to f^* being also strong closed):

$$f_!(\mathcal{H} \underset{W}{\otimes} f^*(\mathcal{H}'))_\bullet \;\simeq\; f_!(\mathcal{H}) \underset{W'}{\otimes} \mathcal{H}' \tag{179}$$

(f) Stability: Over finite classical types $f_!$ and f_* agree to make an ambidextrous adjunction:

$$W \,:\, \mathrm{ClaType}^{\mathrm{fin}} \qquad \vdash \qquad f_! \,\simeq\, f_* \;\;:\;\; \mathrm{QuType}_W \to \mathrm{QuType}\,. \tag{180}$$

Proposition 2.19 (Linear bundle types satisfy Motivic Yoga). *The indexed category $W \mapsto \mathrm{QuType}_w$ of Def. 2.1 satisfies the motivic yoga (Def. 2.18) with respect to the fiberwise tensor product:*

$$\begin{tikzcd}
\mathrm{QuType}_W \times \mathrm{QuType}_W \arrow[rr, "\underset{W}{\otimes}"] & & \mathrm{QuType}_W \\
\left(\begin{bmatrix} \mathcal{H}_\bullet \\ \downarrow \\ W \end{bmatrix}, \begin{bmatrix} \mathcal{H}'_\bullet \\ \downarrow \\ W \end{bmatrix}\right) \arrow[rr, mapsto] & & \begin{bmatrix} \mathcal{H}_w \otimes \mathcal{H}'_w \\ \downarrow \\ (w : W) \end{bmatrix}
\end{tikzcd}$$

Proof. This is straightforward to check. Details for this case and its higher generalization are spelled out in [SS23-EoS, §3.3]. ◻

Remark 2.20 (Modalities via mortivic yoga). We may alternatively see the monoidality of $\triangleright$ and Q just using the motivic yoga (Def. 2.18). For this purpose we shall denote the projection maps involved in a cartesian product as follows:

$$\begin{tikzcd}
 & W \times W' \arrow[dl, "\mathrm{pr}_W"'] \arrow[dr, "\mathrm{pr}_{W'}"] \arrow[dd, "p_{W \times W'}" description] & \\
W \arrow[dr, "p_W"'] & & W' \arrow[dl, "p_{W'}"] \\
 & * &
\end{tikzcd} \tag{181}$$

$$\begin{aligned}
&\triangleright(\mathcal{E} \otimes \mathcal{E}') \\
&= (p_{B \times B'})_!\big((\mathrm{pr}_B)^* \mathcal{E} \otimes (\mathrm{pr}_{B'})^* \mathcal{E}'\big) && \text{def} \\
&\simeq (p_B)_! (\mathrm{pr}_B)_! \big((\mathrm{pr}_B)^* \mathcal{E} \otimes (\mathrm{pr}_{B'})^* \mathcal{E}'\big) && (181) \\
&\simeq (p_B)_! \Big(\mathcal{E} \otimes \big((\mathrm{pr}_{B'})_! (\mathrm{pr}_{B'})^* E\big)\Big) && (179) \\
&\simeq (p_B)_! \Big(\mathcal{E} \otimes \big((p_B)^* (p_{B'})_! E\big)\Big) && (178) \\
&\simeq \big((p_B)_! E\big) \otimes \big((p_{B'})_! \mathcal{E}'\big) && (179)
\end{aligned}$$

$$\begin{aligned}
\mathrm{Q}(B \times B) &= (p_{B \times B})_! (p_{B \times B})^* \mathbb{1} && \text{def} \\
&\simeq (p_{B'})_! (\mathrm{pr}_{B'})_! (\mathrm{pr}_B)^* (p_B)^* \mathbb{1} && (181) \\
&\simeq (p_{B'})_! (p_{B'})^* (p_B)_! (p_B)^* \mathbb{1} && (178) \\
&\simeq (p_{B'})_! \big(\mathbb{1}_{B'} \otimes (p_{B'})^* (p_B)_! (p_B)^* \mathbb{1}\big) && \text{unit law} \\
&\simeq \big((p_{B'})_! \mathbb{1}_{B'}\big) \otimes \big((p_B)_! (p_B)^* \mathbb{1}\big) && (179) \\
&\simeq \big((p_{B'})_! (p_{B'})^* \mathbb{1}\big) \otimes \big((p_B)_! (p_B)^* \mathbb{1}\big) && (177) \\
&= (\mathrm{Q}B) \otimes (\mathrm{Q}B') && \text{def}\,.
\end{aligned}$$

2.2 Classical Epistemic Logic

We lay out our perspective (following [*n*Lab14][Cor20, Ch. 4]) on (S5 Kripke semantics for) modal logic/type theory (Lit. 1.13). This is naturally realized (see Rem. 2.24 below) by *dependent* type theory (Lit. 1.4), with "possible worlds" given by terms of base types and with modal operators given by the (co)monads induced by dependent (co)product[31] type formation followed by context re-extension. The discussion prepares the ground for our formal quantum epistemic logic in §2.3.

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:

$$\begin{tikzcd}
 & W \arrow[rr, "p_w"] & & \Gamma & \\
\text{W-dependent types} & \mathrm{ClaType}_W \arrow[rr, shift left=14pt, "\overset{\text{dependent co-product}}{\coprod_W}"] \arrow[rr, shift right=14pt, "\underset{\text{dependent product}}{\prod_W}"'] \arrow[rr, phantom, shift left=7pt, "\perp"] \arrow[rr, phantom, shift right=7pt, "\perp"] & & \mathrm{ClaType}_\Gamma \arrow[ll, "\text{context } \times W \text{ extension}" description] & \text{types in default context}
\end{tikzcd} \tag{182}$$

via

$$\begin{tikzcd}
D : \mathrm{Type}_W & \vdash & \coprod_W D : & \Gamma \arrow[r] & \mathrm{ClaType} \\
 & & & \gamma \arrow[r, mapsto] & \underset{w : \mathrm{fib}_\gamma(p_W)}{\coprod} D_w \\
D : \mathrm{Type}_\Gamma & \vdash & D \times W : & W \arrow[r] & \mathrm{ClaType} \\
 & & & w \arrow[r, mapsto] & D_{p_W(w)} \\
D : \mathrm{Type}_W & \vdash & \prod_W D : & \Gamma \arrow[r] & \mathrm{ClaType} \\
 & & & \gamma \arrow[r, mapsto] & \underset{w : \mathrm{fib}_\gamma(p_W)}{\prod} D_w
\end{tikzcd} \tag{183}$$

whose (co)restriction along

$$\begin{tikzcd}
\text{types} & \mathrm{ClaType}_\Gamma \arrow[rr, shift left=5pt, "\overset{\text{propositional truncation}}{[-]_0}"] \arrow[rr, phantom, "\perp"] & & \mathrm{Prop}_\Gamma \arrow[ll, shift left=5pt, hook'] & \text{propositions}
\end{tikzcd} \tag{184}$$

gives the quantifiers of first-order logic:

$$\begin{tikzcd}
\text{W-dependent propositions} & \mathrm{Prop}_W \arrow[rr, shift left=14pt, "\overset{\text{existential quantification}}{\exists_W = \left[\coprod_W(-)\right]_0}"] \arrow[rr, shift right=14pt, "\underset{\text{universal quantification}}{\forall_W = \prod_W}"'] \arrow[rr, phantom, shift left=7pt, "\perp"] \arrow[rr, phantom, shift right=7pt, "\perp"] & & \mathrm{Prop}_\Gamma \arrow[ll, "\text{context } \times W \text{ extension}" description] & \text{propositions in default context}
\end{tikzcd} \tag{185}$$

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

---

[31] 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.

composing the adjoint type constructors (182) to endo-functors yields a pair of adjoint pairs of (co)monads:

$$\begin{tikzcd}
& W \arrow[rr, "p_W"] && \Gamma & \\
\text{actual data} \arrow[r, phantom, "\perp"] & \mathrm{ClaType}_W \arrow[loop above, "\overset{\text{possibly}}{\Diamond_W}"] \arrow[loop below, "\underset{\text{necessarily}}{\Box_W}"'] \arrow[rr, shift left=12pt, "\overset{\text{dependent co-product}}{\coprod_W}"] \arrow[rr, phantom, shift left=6pt, "\perp"] \arrow[rr, phantom, shift right=6pt, "\perp"] \arrow[rr, shift right=12pt, "\underset{\text{dependent product}}{\prod_W}"'] && \mathrm{ClaType}_\Gamma \arrow[ll, "(-)\times W"] \arrow[loop above, "\overset{\text{randomly}}{\text{☆}_W}"] \arrow[loop below, "\underset{\text{indefinitely}}{\bigcirc_W}"'] \arrow[r, phantom, "\perp"] & \text{potential data}
\end{tikzcd}$$ (186)

whose (co)restriction along propositional truncation (184) we shall denote by the same symbols:

$$\begin{tikzcd}
& W \arrow[rr, "p_W"] && \Gamma & \\
\text{actual propositions} \arrow[r, phantom, "\perp"] & \mathrm{Prop}_W \arrow[loop above, "\overset{\text{possibly}}{\Diamond_W}"] \arrow[loop below, "\underset{\text{necessarily}}{\Box_W}"'] \arrow[rr, shift left=12pt, "\overset{\substack{\text{0-truncated} \\ \text{dependent co-product}}}{\left[\coprod_W(-)\right]_0}"] \arrow[rr, phantom, shift left=6pt, "\perp"] \arrow[rr, phantom, shift right=6pt, "\perp"] \arrow[rr, shift right=12pt, "\underset{\text{dependent product}}{\prod_W}"'] && \mathrm{Prop}_\Gamma \arrow[ll, "(-)\times W"] \arrow[loop above, "\overset{\text{randomly}}{\text{☆}_W}"] \arrow[loop below, "\underset{\text{indefinitely}}{\bigcirc_W}"'] \arrow[r, phantom, "\perp"] & \text{potential propositions}
\end{tikzcd}$$ (187)

Actuality logic. The terminology on the left of diagram(186) is justified by the following Remark 2.21 and the observation of Theorem 2.23 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 2.21 (**Epistemic interpretation of dependent types).** Concretely, we may read these modal operators (186) as follows, where we use the traditional language of "possible worlds" (Lit. 1.13) 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. 1.2).
(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:	$\Diamond_W P_\bullet$ means:
"*P does* or *is known to* hold **necessarily**" namely, no matter which world w is measured.	"*P does* or *is known* to hold **actually**" namely for the *given* world w measured.	"*P does* or *is known* to hold **possibly**" namely for *some* possibly measured world w.

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

$$\begin{tikzcd}
& \text{necessarily } D_\bullet & \text{entails} & \text{actually } D_\bullet & \text{entails} & \text{possibly } D_\bullet \\
& \Box_W D_\bullet \arrow[rr, "\mathrm{obt}^{\Box_W}_{D_\bullet}"] && D_\bullet \arrow[rr, "\mathrm{ret}^{\Diamond_W}_{D_\bullet}"] && \Diamond_W D_\bullet \\
w : W \quad \vdash & \prod\limits_{w':W} D_{w'} \arrow[rr, "(d_{w':W}) \,\mapsto\, d_w"] && D_w \arrow[rr, "d_w \,\mapsto\, (w, d_w)"] && \coprod\limits_{w':W} D_{w'}
\end{tikzcd}$$ (188)

Remark 2.22 (**Hexagon of epistemic entailments).** The *naturality* of the transformations (188) is reflected in commuting squares as shown in the following diagram (189), whose hexagonal composition gives the diagram (7) announced in the Introduction (there evaluated for linear/quantum types, which we come to in §2.3, but the existence of the commuting hexagon as such depends only on the naturality of the epistemic entailments):

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

$$\begin{tikzcd}
 & \square\lozenge D_\bullet \arrow[rr, "\square\lozenge G_\bullet"] \arrow[d, phantom, "\square(\mathrm{ret}^{\lozenge}_{G_\bullet})"] & & \square\lozenge D'_\bullet \arrow[dr, "\mathrm{obt}^{\square}_{\lozenge D'_\bullet}"] & \\
\square D_\bullet \arrow[ur, "\square(\mathrm{ret}^{\lozenge}_{D_\bullet})"] \arrow[rr, "\square G_\bullet"] \arrow[dr, "\mathrm{obt}^{\square}_{D_\bullet}"'] & \mathrm{obt}^{\square}_{G_\bullet} & \square D'_\bullet \arrow[ur, "\square(\mathrm{ret}^{\lozenge}_{D'_\bullet})"] \arrow[dr, "\mathrm{obt}^{\square}_{D'_\bullet}"'] & \mathrm{obt}^{\square}_{(\mathrm{ret}^{\lozenge}_{D'_\bullet})} & \lozenge D'_\bullet \\
 & D_\bullet \arrow[rr, "G_\bullet"] & & D'_\bullet \arrow[ur, "\mathrm{ret}^{\lozenge}_{D'_\bullet}"'] &
\end{tikzcd}$$

$$\| \tag{189}$$

$$\begin{tikzcd}
 & \square\lozenge D_\bullet \arrow[rr, "\square\lozenge G_\bullet"] \arrow[dr, "\mathrm{obt}^{\square}_{\lozenge D_\bullet}"] & \mathrm{obt}^{\square}_{\lozenge G_\bullet} & \square\lozenge D'_\bullet \arrow[dr, "\mathrm{obt}^{\square}_{\lozenge D'_\bullet}"] & \\
\square D_\bullet \arrow[ur, "\square(\mathrm{ret}^{\lozenge}_{D_\bullet})"] \arrow[dr, "\mathrm{obt}^{\square}_{D_\bullet}"'] & \mathrm{obt}^{\square}_{(\mathrm{ret}^{\lozenge}_{D_\bullet})} & \lozenge D_\bullet \arrow[rr, "\lozenge G_\bullet"] & \mathrm{ret}^{\lozenge}_{G_\bullet} & \lozenge D'_\bullet \\
 & D_\bullet \arrow[rr, "G_\bullet"] \arrow[ur, "\mathrm{ret}^{\lozenge}_{D_\bullet}"'] & & D'_\bullet \arrow[ur, "\mathrm{ret}^{\lozenge}_{D'_\bullet}"'] &
\end{tikzcd}$$

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

Theorem 2.23 (S5 Kripke semantics as co-monadic descent). *The possible-worlds Kripke semantics* (63) *for S5 modal logic are precisely given by dependent type formation* (186) *(for* $\mathrm{ClaType} \equiv \mathrm{Set}$*) where a Kripke frame* ($W : \mathrm{Set},\ R : W{\times}W \to \mathrm{Prop}$) *corresponds to that display map* (182) *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 (183), the induced comonad (186) acts as

$$P : \mathrm{Prop}_W \quad \vdash \quad \begin{array}{rcl} \square_W P \;:\; W & \longrightarrow & \mathrm{Prop} \\ w & \mapsto & \underset{w' \,:\, \mathrm{fib}_{p_W(w)}(p_W)}{\forall} P(w') \end{array} \tag{190}$$

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 (190) equals the traditional formula (63) for the Kripke semantics of the modal operator. □

Remark 2.24 (Dependent type theory as universal Epistemic modal type theory).
(i) Thm. 2.23 suggests that one may regard dependent type theory equivalently as a universal form of epistemic type theory (Lit. 1.14) in generalization of how modal logic 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 (182)(185) to the associated adjoint pairs of (co)monads (186)(187). (An analogous change in perspective happens in (algebraic) geometry when expressing *descent theory* in terms of *monadic descent*.)
(ii) 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 §2.3) 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 (186) are known in effectful classical computer science (Lit. 1.17) as the *W-(co)reader (co)monad*, (120) often denoted as on the right here:

$$\begin{aligned} \bigcirc_W D \;&\equiv\; [W,\, D] && \textbf{W-reader monad} \\ \text{☆}_W D \;&\equiv\; W \times D && \textbf{W-coreader comonad} \end{aligned} \tag{191}$$

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 (186) 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

$$\begin{array}{ccccc} \mathrm{ClaType}_\Gamma & \xleftrightarrow{\qquad\qquad\qquad \sim \qquad\qquad\qquad} & & & \mathrm{ClaType}_W^{\Diamond_W} \\ D : \mathrm{Type}_\Gamma & \xleftrightarrow{\;\;\sim\;\;} & \big(D_\bullet : \mathrm{Type}_W,\; \rho \,:\, \Diamond_W D_\bullet \longrightarrow D_\bullet, & \mathrm{utl}_{\Diamond_W}(\rho), & \mathrm{act}_{\Diamond_W}(\rho)\big) \\ \textbf{potential data} & \textbf{is equivalently} & \textbf{data whose} \quad \textbf{possibility entails its actuality,} & & \textbf{consistently} \end{array} \tag{192}$$

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

Proposition 2.25 (Potential data as possibility modal data). *For $p_W : W \twoheadrightarrow \Gamma$ an epimorphism (as in Thm. 2.23), the context extension* $(\text{-}) \times W : \mathrm{ClaType}_\Gamma \to \mathrm{ClaType}_W$ *is monadic* (98) *whence the potential types* (186) *are identified with the (free) possibility-modal types* (93) *and hence* (121) *also with the necessity-modal types:*

$$\begin{tikzcd}
 & \text{possibly} & {} \arrow[r, "\coprod_W"] \arrow[d, phantom, "\perp"] & \mathrm{ClaType}_W^{\Diamond_W} \arrow[d, phantom, "\wr"] & \textbf{possibility modal data} \\
\textbf{actual data} \perp & \mathrm{ClaType}_W \arrow[loop above, "\Diamond_W"] \arrow[loop below, "\Box_W"'] & {} \arrow[l] & \mathrm{ClaType}_\Gamma \arrow[l, dash, "\times W" description] & \textbf{potential data} \\
 & \text{necessarily} & {} \arrow[r, "\prod_W"'] \arrow[u, phantom, "\perp"] & \mathrm{ClaType}_W^{\Box_W} \arrow[u, phantom, "\wr"] & \textbf{necessity modal data}
\end{tikzcd} \tag{193}$$

Proof. By the Monadicity Theorem (98) 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 $\gamma : \Gamma$. □

Remark 2.26 (Relation to monadic descent). The statement and proof of Prop. 2.25 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 2.27 (Relation to lenses). In the case Type = Set, the statement of Prop. 2.25 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"):

$$\left(\begin{tikzcd}[column sep=small]
\textbf{slice object} & & & \textbf{$\Diamond_W$-modale structure} & & & \textbf{$\Diamond_W$-unit law} & & \textbf{$\Diamond_W$-action property} & \\
S \arrow[d, "\mathrm{get}"] & & S \times W \arrow[rr, "\mathrm{put}"] \arrow[dr, "\mathrm{pr}_W"'] & & S \arrow[dl, "\mathrm{get}"] & & W \times S \arrow[dr, "\mathrm{put}"] & & W \times W \times S \arrow[r, "\mathrm{pr}_1 \times \mathrm{pr}_3"] \arrow[d, "\mathrm{id}_W \times \mathrm{put}"'] & W \times S \arrow[d, "\mathrm{put}"] \\
W & \in \mathrm{Type}_W & & W & & S \arrow[ur, "\mathrm{get}\times\mathrm{id}"] \arrow[rr, equal] & & S & W \times S \arrow[r, "\mathrm{put}"] & S \\
\textbf{database type S with} & & & \textbf{W-write functionality} & & & \textbf{overwriting identi-} & & \textbf{subsequent writing} & \\
\textbf{W-read functionality} & & & \textbf{verified by W-reading} & & & \textbf{cally has no effect} & & \textbf{overwrites previous} &
\end{tikzcd}\right) : (\mathrm{Type}_W)^{\Diamond_W} \tag{194}$$

and the upshot of the monadicity statement (Prop. 2.25, [JRW10, Thm. 12][32]) 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 = pr_W, etc.

---

[32] [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.

Random and (in)definite data. The (co)monads $\bigcirc$ ($\star$) on the right of (186) are well-known in terms of (co)effects in computer science (Lit. 1.17) as the "(co)reader (co)monad" (120), 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-data d in a *definite* but *random* world w (as in "random access")	D is the type of plain D-data d only *potentially* in some possible world	$\bigcirc_W P_\bullet$ is the type of: *indefinite* D-data $w \mapsto d_w$ contingent on a pending choice of possible world w.

$$\begin{tikzcd}
\textbf{randomly } P & \textbf{entails} & \textbf{potentially } P & \textbf{entails} & \textbf{indefinitely } P \\
\star_W P \arrow[rr, "\mathrm{ret}_P^{\star_W}"] & & P \arrow[rr, "\mathrm{obt}_P^{\bigcirc_W}"] & & \bigcirc_W P \\
\coprod\limits_{w':W} P \arrow[rr, "(w,p) \mapsto p"] & & P \arrow[rr, "p \mapsto (w' \mapsto p)"] & & \prod\limits_{w':W} P
\end{tikzcd} \tag{195}$$

In particular, the monadic effect model (cf. Lit. 1.17) for operating on the parameter space W as on a *random access memory* (RAM) is the state monad (83), which we may realize as the composite

$$\underset{W}{\bigcirc}\,\underset{W}{\star}\, D \;\simeq\; \prod_{W}\coprod_{W} D \;\simeq\; [W,\, W \times D] \;\equiv\; W\mathrm{State}(D), \qquad \begin{tikzcd} \mathrm{Type} \arrow[loop left, "W\mathrm{State}"] \arrow[r, shift left=2, "\star_W"] \arrow[r, phantom, "\perp"] & \mathrm{Type} \arrow[l, shift left=2, "\bigcirc_W"] \end{tikzcd} . \tag{196}$$

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:

$$\begin{tikzcd}
\textbf{actual data} \arrow[r, phantom, "\perp"] & \mathrm{ClaType}_W \arrow[loop above, "\Diamond_W", "\textbf{possibly}"'] \arrow[loop below, "\Box_W"', "\textbf{necessarily}"] \arrow[rr, shift left=6, "\coprod_W", "\textbf{dependent co-product}"'] \arrow[rr, phantom, shift left=3, "\perp"] \arrow[rr, phantom, shift right=3, "\perp"] \arrow[rr, shift right=6, "\prod_W"', "\textbf{dependent product}"] & & \mathrm{ClaType} \arrow[ll, "\times W"'] \arrow[loop above, "\star_W", "\textbf{randomly}"'] \arrow[loop below, "\bigcirc_W"', "\textbf{indefinitely}"] \arrow[r, phantom, "\perp"] & \textbf{potential data}
\end{tikzcd}$$

$$\begin{tikzcd}
& \textbf{necessarily } P_\bullet & \textbf{entails} & \textbf{actually } P_\bullet & \textbf{entails} & \textbf{possibly } P_\bullet \\
& \Box_W\, P_\bullet \arrow[rr, "\epsilon_{P_\bullet}^{\Box_W}"] & & P_\bullet \arrow[rr, "\eta_{P_\bullet}^{\Diamond_W}"] & & \Diamond_W\, P_\bullet \\
w : W \;\vdash & \prod\limits_{w':W} P_{w'} \arrow[rr, "(w' \mapsto p_{w'}) \;\mapsto\; p_w"] & & P_w \arrow[rr, "p_w \;\mapsto\; (w,\, p_w)"] & & \coprod\limits_{w':W} P_{w'} \\
& \textbf{randomly } P & \textbf{entails} & \textbf{potentially } P & \textbf{entails} & \textbf{indefinitely } P \\
& \star_W P \arrow[rr, "\mathrm{ret}_P^{\star_W}"] & & P \arrow[rr, "\mathrm{obt}_P^{\bigcirc_W}"] & & \bigcirc_W P \\
& \coprod\limits_{w':W} P \arrow[rr, "(w,p) \mapsto p"] & & P \arrow[rr, "p \mapsto (w' \mapsto p)"] & & \prod\limits_{w':W} P
\end{tikzcd} \tag{197}$$

Next we proceed to find the quantum analog (202) of this logic.

2.3 Quantum Epistemic Logic

On the backdrop (§2.2) of classical (intuitionistic) epistemic type theory understood as an equivalent re-interpretation of classical (intuitionistic) dependent type theory, and in view (§2.1) 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 (171) of dependent *linear* type formation

$$\begin{tikzcd}[column sep=large, row sep=huge]
\begin{array}{c}\textbf{Finite classical context}\\\textbf{(parameters, measurement}\\\textbf{results: “possible worlds”)}\end{array} & W \arrow[rr, "\overset{\text{display map}}{p_W}", "\text{dependent co-product}"'] & & \Gamma & \begin{array}{c}\textbf{Reference context}\\\textbf{(classes of “worlds”)}\end{array} & \\
\begin{array}{c}\textbf{Classical type system}\\\textbf{dependent on context}\end{array} & (\mathrm{ClaType}_W, \overset{\text{product}}{\times_W}) \arrow[rr, shift left=3ex, "\coprod_W"] \arrow[rr, shift left=1.5ex, phantom, "\perp"] \arrow[rr, shift right=1.5ex, phantom, "\perp"] \arrow[rr, shift right=3ex, "\prod_W"', "\text{dependent product}"'{yshift=-3ex}] & & (\mathrm{ClaType}_\Gamma, \overset{\text{product}}{\times_\Gamma}) \arrow[ll, "\times W"'] & \begin{array}{c}\textbf{Classical (intuit.)}\\\textbf{type system}\end{array} & \begin{array}{c}\text{Classical base change /}\\\text{classical quantification}\end{array} \\
\begin{array}{c}\textbf{Linear type system}\\\textbf{in classical context}\end{array} & (\mathrm{QuType}_W, \overset{\text{tensor}}{\otimes_W}) \arrow[rr, shift left=3ex, "\oplus_W", "\text{dependent direct sum}"{yshift=3ex}] \arrow[rr, shift left=1.5ex, phantom, "\perp"] \arrow[rr, shift right=1.5ex, phantom, "\perp"] \arrow[rr, shift right=3ex, "\oplus_W"'] & & (\mathrm{QuType}_\Gamma, \overset{\text{tensor}}{\otimes_\Gamma}) \arrow[ll, "\otimes \mathbb{1}_W"'] & \begin{array}{c}\textbf{Linear}\\\textbf{type system}\end{array} & \begin{array}{c}\text{Quantum base change}\\\text{/ Motivic Yoga}\end{array}
\end{tikzcd}$$

by passing to the induced (co)monads (74), which we denote by the same symbols as their classical counterparts (186):

$$\begin{tikzcd}[column sep=huge]
& W \arrow[rr, "p_W"] & & \Gamma & \\
\textbf{Actual quantum data} \quad \perp & \mathrm{QuType}_W \arrow[loop above, "\overset{\textbf{possibly}}{\Diamond_W}"] \arrow[loop below, "\underset{\textbf{necessarily}}{\Box_W}"'] \arrow[rr, shift left=3ex, "\overset{\textbf{dependent direct sum}}{\oplus_W}"] \arrow[rr, shift left=1.5ex, phantom, "\perp"] \arrow[rr, shift right=1.5ex, phantom, "\perp"] \arrow[rr, shift right=3ex, "\underset{\textbf{dependent direct sum}}{\oplus_W}"'] & & \mathrm{QuType}_\Gamma \arrow[ll, "\otimes \mathbb{1}_W"'] \arrow[loop above, "\overset{\textbf{randomly}}{\star_W}"] \arrow[loop below, "\underset{\textbf{indefinitely}}{\bigcirc_W}"'] & \perp \quad \textbf{Potential quantum data}
\end{tikzcd} \tag{198}$$

A key point here is the *ambitexterity* (171) of the base change for dependent linear types along a finite classical type W:

$$W \,:\, \mathrm{ClaType}^{\mathrm{fin}} \qquad \vdash \qquad \Big(\bigoplus_W \;\dashv\; \otimes \mathbb{1}_W \;\dashv\; \bigoplus_W \Big) \tag{199}$$

It is now as elementary to work out (this is the next Prop. 2.29) the (co)units of these (co)monads as it is interesting, in view of quantum epistemology (Lit. 1.1).

Example 2.28 (Quantum state collapse from quantum modality). The quantum $\Box$-counit is analyzed as follows:
Consider, for simplicity, a quantum type independent of worlds b : Bit. $\quad \mathcal{H} \;\in\; \mathrm{QuType}_* \xrightarrow{\otimes \mathbb{1}_{\mathrm{Bit}}} \mathrm{QuType}_{\mathrm{Bit}}$

Observe that $\quad \Box_{\mathrm{Bit}} \mathcal{H} \;\equiv\; \oplus_B \mathcal{H} \;\simeq\; \mathcal{H} \otimes \mathrm{QBit} \quad \in \mathrm{QuType}_* \xrightarrow{\otimes \mathbb{1}_{\mathrm{Bit}}} \mathrm{QuType}_{\mathrm{Bit}}\,,$
arising as the *limiting cone* over the Bit-indexed diagram constant on $\mathcal{H}$.

Hence the $\Box$-counit map is over b : Bit the projection onto the bth component.

$$b : B \quad \vdash \quad \begin{tikzcd}
(\Box_{\mathrm{Bit}} \mathcal{H})_b \arrow[r, "(\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathcal{H}})_b"] \arrow[d, phantom, "\shortparallel"] & \mathcal{H}_b \arrow[d, phantom, "\shortparallel"] \\
\mathcal{H} \otimes \mathrm{QBit} \arrow[r, "\sum_{b'} |\psi_{b'}\rangle \otimes |b'\rangle \longmapsto |\psi_b\rangle"] & \mathcal{H}\,.
\end{tikzcd}$$

But this reflects the quantum measurement process:
Observing classical outcome b, the quantum state is collapsed onto the subspace spanned by $|b\rangle$.

Proceeding in this fashion, one finds:

Proposition 2.29 (Component expressions of the quantum (co)monad (co)units). *The (co)units and (co)joins of the (co)monads in* (198) *are given, in components, as follows:*

Epistemic entailments in Quantum modal logic

$$\begin{tikzcd} \underset{W}{\Box}\mathcal{H}_\bullet \arrow[rr, "\mathrm{obt}^{\Box_W}_{\mathcal{H}_\bullet}", "\text{necessity counit}"'] & & \mathcal{H}_\bullet \\ w : W \vdash \bigoplus_{w'}\mathcal{H}_w \arrow[rr, "\oplus_{w'}|\psi_{w'}\rangle \;\mapsto\; |\psi_w\rangle", "\text{quantum state collapse}"'] & & \mathcal{H}_w \end{tikzcd}$$

" *what is necessary is actualized* "

$$\begin{tikzcd} \mathcal{H}_\bullet \arrow[rr, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}", "\text{possibility unit}"'] & & \underset{W}{\Diamond}\mathcal{H}_\bullet \\ w : W \vdash \mathcal{H}_w \arrow[rr, "|\psi_w\rangle \;\mapsto\; \oplus_{w'}\delta^{w'}_w|\psi_w\rangle", "\text{quantum state preparation}"'] & & \bigoplus_{w'}\mathcal{H}_{w'} \end{tikzcd}$$

" *what is actual is possible* "

" *what is random exists potentially* "

$$\begin{tikzcd} \underset{W}{\text{☆}}\mathcal{H} \arrow[rr, hook, "\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}}", "\text{randomness counit}"'] & & \mathcal{H} \\ \bigoplus_{w}\mathcal{H} \arrow[rr, "\oplus_w|\psi_w\rangle \;\mapsto\; \sum_w|\psi_w\rangle", "\text{quantum superposition}"'] & & \mathcal{H} \end{tikzcd}$$

" *what exists potentially is indeterminate* "

$$\begin{tikzcd} \mathcal{H} \arrow[rr, "\mathrm{ret}^{\bigcirc_B}_{\mathcal{H}}", "\text{indefiniteness unit}"'] & & \underset{W}{\bigcirc}\mathcal{H} \\ \mathcal{H} \arrow[rr, "|\psi\rangle \;\mapsto\; \oplus_w|\psi\rangle", "\text{quantum parallelism}"'] & & \bigoplus_{W}\mathcal{H} \end{tikzcd}$$

(200)

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

$$\begin{tikzcd} \underset{W}{\text{☆}}\mathcal{H} \arrow[rr, "\mathrm{dplc}^{\text{☆}_W}_{\mathcal{H}}", "\text{randomness cojoin}"'] & & \underset{W}{\text{☆}}\underset{W}{\text{☆}}\mathcal{H} \\ \bigoplus_{w}\Big(\mathcal{H} \arrow[rr, "\mathrm{obt}^{\Diamond_W}_{\mathcal{H}}", "\text{quantum state prepar.}"'] & & \underset{W}{\Diamond}\mathcal{H}\Big) \\ |\psi_w\rangle & \mapsto & \oplus_{w'}\delta^{w'}_w|\psi_w\rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{WW}{\Diamond\Diamond}\mathcal{H}_\bullet \arrow[rr, "\mathrm{join}^{\Diamond_W}_{\mathcal{H}_\bullet}", "\text{possibility join}"'] & & \underset{W}{\Diamond}\mathcal{H}_\bullet \\ w : W \vdash \underset{W}{\text{☆}}\bigoplus_{W}\mathcal{H}_\bullet \arrow[rr, "\mathrm{obt}^{\text{☆}_W}_{\oplus_W\mathcal{H}_\bullet}", "\text{quantum superposition}"'] & & \bigoplus_{W}\mathcal{H}_\bullet \\ \bigoplus_{w''}|\psi_{w,w',w''}\rangle & \mapsto & \sum_{w''}|\psi_{w,w',w''}\rangle \end{tikzcd}$$

$$\begin{tikzcd} \underset{W}{\Box}\mathcal{H}_\bullet \arrow[rr, "\mathrm{dplc}^{\Box_W}_{\mathcal{H}_\bullet}", "\text{necessity cojoin}"'] & & \underset{W}{\Box}\underset{W}{\Box}\mathcal{H}_\bullet \\ w : A \vdash \bigoplus_{W}\mathcal{H}_\bullet \arrow[rr, "\mathrm{ret}^{\bigcirc_W}_{\oplus_W\mathcal{H}_\bullet}", "\text{quantum parallelism}"'] & & \underset{W}{\bigcirc}\bigoplus_{W}\mathcal{H}_\bullet \\ |\psi_{w,w'}\rangle & \mapsto & \bigoplus_{w''}|\psi_{w,w'}\rangle \end{tikzcd}$$

Here the (co)joins in the lower half follow from the (co)units in the top half, via (76).

Monadicity of quantum data. We observe that quantum data as in (198) is characterized by two monadicity theorems:

- Prop. 2.30: Potential quantum data is possibility-modal actual data.
- Prop. 2.32: Actual quantum data is indefiniteness-modal potential data.

First, we have the following quantum analog of the classical situation from Prop. 2.25:

Proposition 2.30 (Potential quantum data as possibility-modal actual data). *For* $p_W : W \twoheadrightarrow \Gamma$ *an epimorphism (as in Thm. 2.23) the context extension* $(\text{-}) \otimes \mathbb{1}_w : \mathrm{QuType}_\Gamma \to \mathrm{QuType}_w$ *is monadic* (98) *whence the potential quantum types* (198) *are identified with the (free) possibility/necessity modal types* (93) *(just as classically* (193)*):*

$$\begin{tikzcd} & \overset{\text{possibly}}{\Diamond_W} \arrow[dr, bend left] & & & \mathrm{QuType}^{\Diamond_W}_W & \text{Possibility modal data} \\ \text{Actual quantum data} & \perp & \mathrm{QuType}_W \arrow[ul, dash] \arrow[dl, dash] \arrow[urr, "\bigoplus_W"] \arrow[drr, "\bigoplus_W"'] \arrow[rr, phantom, shift left=3, "\perp"] \arrow[rr, phantom, shift right=3, "\perp"] & & \mathrm{QuType}_\Gamma \arrow[ll, "\otimes\mathbb{1}_W"'] \arrow[u, phantom, "\wr"] \arrow[d, phantom, "\wr"] & \text{Potential quantum data} \\ & \underset{\text{necessarily}}{\Box_W} \arrow[ur, bend right] & & & \mathrm{QuType}^{\Box_W}_W & \text{Necessity modal data} \end{tikzcd}$$

(201)

Proof. This statement has verbatim the same abstract proof – via the monadicity theorem (99) and the comparison statement (121) – as its classical counterpart in Prop. 2.25, 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 2.31 (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:

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

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

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

In summary so far, we have found a quantum epistemic logic with the following interpretations, analogous to (197):

$$\begin{tikzcd}[column sep=small]
& & & & & & \text{principle of quantum compulsion:} & & \\
& & \textbf{necessarily } \mathcal{H}_\bullet & \textbf{entails} & \textbf{actually } \mathcal{H}_\bullet & \textbf{entails} & \textbf{possibly } \mathcal{H}_\bullet & \textbf{is} & \textbf{necessarily } \mathcal{H}_\bullet \\
& & \Box_W\,\mathcal{H}_\bullet \arrow[rr, "\mathrm{obt}^{\Box_W}_{\mathcal{H}_\bullet}"] & & \mathcal{H}_\bullet \arrow[rr, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}_\bullet}"] & & \Diamond_W\,\mathcal{H}_\bullet & \simeq & \Box_W\,\mathcal{H}_\bullet \\
\textbf{In world } w : W & \textbf{observe...}\ \vdash & \mathcal{H} \arrow[rr, two heads, "\oplus_{w'}|\psi_{w'}\rangle\ \mapsto\ |\psi_w\rangle", "\textbf{measurement collapse}"'] \arrow[rrrr, bend right, "\textbf{linear projector onto sub-Hilbert space } \mathcal{H}_w"'] & & \mathcal{H}_w \arrow[rr, hook, "|\psi_w\rangle \mapsto \oplus_{w'}\delta^{w'}_w|\psi_w\rangle", "\textbf{state preparation}"'] & & \mathcal{H}, & & \text{where } \mathcal{H} \;:=\; \bigoplus\limits_{w':W}\mathcal{H}_{w'} \\
& & \textbf{randomly } \mathcal{H} & \textbf{entails} & \textbf{potentially } \mathcal{H} & \textbf{entails} & \textbf{indefinitely } \mathcal{H} & & \\
& & \star_W\,\mathcal{H} \arrow[rr, "\mathrm{obt}^{\star_W}_{\mathcal{H}}"] & & \mathcal{H} \arrow[rr, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}"] & & \bigcirc_W\,\mathcal{H} & & \\
& & \bigoplus\limits_{w:W}\mathcal{H} \arrow[rr, "\oplus_W|\psi_W\rangle \mapsto \Sigma_W|\psi_W\rangle", "\textbf{quantum superposition}"'] & & \mathcal{H} \arrow[rr, "|\psi\rangle \mapsto \oplus_{w':W}|\psi\rangle", "\textbf{quantum parallelism}"'] & & \bigoplus\limits_{b:B}\mathcal{H} & &
\end{tikzcd} \tag{202}$$

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

Proposition 2.32 (Actual quantum data as indefiniteness-modal potential data). *For $W : \mathrm{ClaType}^{\mathrm{fin}}_\Gamma$ 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:*

$$\begin{tikzcd}
& & & \text{randomly} & \\
\text{Randomness modal data} & \mathrm{QuType}_{\Gamma}^{\star_W} \arrow[rr, "\oplus_W"] \arrow[d, phantom, "\wr"] & & {} & \\
& {} & {} \arrow[d, phantom, "\perp"] & & \\
\text{Actual quantum data} & \mathrm{QuType}_W \arrow[d, phantom, "\wr"] & {} & \mathrm{QuType}_{\Gamma} \arrow[ll, "\otimes \mathbb{1}_W"'] \arrow[loop above, "\star_W"] \arrow[loop below, "\bigcirc_W"'] \arrow[r, phantom, "\perp"] & \text{Potential quantum data} \\
& {} & {} \arrow[u, phantom, "\perp"] & & \\
\text{Indefiniteness modal data} & \mathrm{QuType}_{\Gamma}^{\bigcirc_W} \arrow[rr, "\oplus_W"'] & & {} & \\
& & & \text{indefinitely} &
\end{tikzcd} \tag{203}$$

Proof. Due to ambidexterity (199) for finite W, in the quantum case also $\oplus_W$ is both a left and right adjoint, as shown. Therefore the monadicity theorem (99) 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 $\star$ now follows by (121). ☐

Remark 2.33 (Effective perspective on quantum epistemology)**.** Prop. 2.32 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. 1.17). Hence we shall refer to these equivalent perspectives as the *epistemic* and the *effective* perspective, respectively:

$$\begin{tikzcd}
\textbf{Epistemic perspective} \arrow[dd, leftrightarrow, "\text{monadicity}", "\text{of } \oplus_W"'] & \mathrm{QuType}_W \arrow[dd, shift left=2, "\bigoplus\limits_W"] & \underset{\text{map of w-dependent types}}{\mathcal{H}_\bullet \xrightarrow{\;G_\bullet\;} \mathcal{K}_\bullet} \arrow[dd, leftrightarrow] & \underset{\text{in-dependent type}}{\mathcal{H}} \arrow[dd, leftrightarrow] & \underset{\substack{\text{w-dependent map of}\\ \text{in-dependent types}}}{\mathcal{H} \xrightarrow{\;G_\bullet\;} \mathcal{H}} \arrow[dd, leftrightarrow] \\
& {} & & & \\
\textbf{Effective perspective} & \underset{\big(\underset{W}{\bigcirc}\circlearrowright\big)}{\mathrm{QuType}_{\Gamma}} \arrow[uu, shift left=2, "\otimes \mathbb{1}_W"] \arrow[uu, phantom, "\dashv"] & \overset{\text{homomorphism of $\bigcirc_W$-modales}}{\underset{\big(\underset{W}{\bigcirc}\circlearrowright\big) \qquad\qquad \big(\underset{W}{\bigcirc}\circlearrowright\big)}{\bigoplus\limits_W \mathcal{H}_\bullet \xrightarrow{\;\bigoplus\limits_W G_\bullet\;} \bigoplus\limits_W \mathcal{K}_\bullet}} & \overset{\text{free $\bigcirc_W$-modale}}{\underset{W}{\bigcirc}\mathcal{H}} & \overset{\text{$\bigcirc_W$-Kleisli map}}{\mathrm{bind}\Big(\mathcal{H} \xrightarrow[\bigoplus\limits_W G_\bullet \circ \mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}]{} \underset{W}{\bigcirc}\mathcal{K}\Big)}
\end{tikzcd} \tag{204}$$

The effective perspective on the epistemic entailments (202) yields an effect-language for quantum measurement and controlled quantum gates – this we discuss next in §2.4.

Remark 2.34 (**Relation to `zxCalculus`**)**.** Something close to the identification $(\mathrm{QuType}_\Gamma)^{\star_W} \simeq \mathrm{QuType}_W$ (in Prop. 2.32) has previously been observed in [CPav08, Thm. 1.5] (cf. Lit. 1.18), 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. 1.18). In particular, this shows that/how `LHoTT/QS` can be used for certifying (type-checking) `zxCalculus`-protocols:

Proposition 2.35 (**Quantum (co)effects via Frobenius algebra**)**.**

(i) *For* W : ClaType*, the W-(co)reader (co)monad on linear types (§2.3) is equivalent to the linear version* $\mathrm{Q}W \otimes (\text{-})$ *of the (co)writer (co)monad* (82) *induced by the canonical (co)algebra structure on* $\mathrm{Q}W \equiv \oplus_W \mathbb{1}$*;*

(ii) *If* W : $\mathrm{ClaType}^{\mathrm{fin}}$ *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* $\mathrm{Q}W = \bigoplus\limits_W \mathbb{1}$ *(cf. Lit. 1.18):*

<table>
<tr><th colspan="2">Frobenius structure on $\mathrm{Q}W = \oplus_W \mathbb{1}$</th></tr>
<tr><th>Algebra structure</th><th>Coalgebra structure</th></tr>
<tr><td>$\begin{array}{ccc} \mathbb{1} & \xrightarrow{\mathrm{unit}_{\mathrm{Q}W}} & \mathrm{Q}W \\ 1 & \mapsto & \oplus_w \vert w\rangle \end{array}$</td><td>$\begin{array}{ccc} \mathrm{Q}W & \xrightarrow{\mathrm{counit}_{\mathrm{Q}W}} & \mathbb{1} \\ \vert w\rangle & \mapsto & 1 \end{array}$</td></tr>
<tr><td>$\begin{array}{ccc} \mathrm{Q}W \otimes \mathrm{Q}W & \xrightarrow{\mathrm{prod}_{\mathrm{Q}W}} & \mathrm{Q}W \\ \vert w_1\rangle \otimes \vert w_2\rangle & \mapsto & \delta^{w_2}_{w_1} \vert w_2\rangle \end{array}$</td><td>$\begin{array}{ccc} \mathrm{Q}W & \xrightarrow{\mathrm{coprod}_{\mathrm{Q}W}} & \mathrm{Q}W \otimes \mathrm{Q}W \\ \vert w\rangle & \mapsto & \vert w\rangle \otimes \vert w\rangle \end{array}$</td></tr>
</table>

$$\begin{tikzcd}
\textbf{Quantum indefiniteness} & & & & \textbf{Quantum randomness} \\
\textbf{quantum reader} & & \textbf{quantum (co)writer} & & \textbf{quantum co-reader} \\
\underset{W}{\bigcirc} & \simeq & (\mathrm{Q}W)\mathrm{Write} & \simeq & \underset{W}{\star} \\
\mathrm{Monads} & & \mathrm{FrobMonads} \arrow[ll] \arrow[rr] & & \mathrm{CoMonads}
\end{tikzcd}$$

Proof. With Prop. 2.29, this is a straightforward matter of unwinding the definitions:

Measurement

$$\begin{tikzcd}
\textbf{epistemic} & \underset{W}{\Box}(\underset{W}{\mathbb{1}}\otimes\mathcal{H}) \arrow[r, "\mathrm{obt}^{\Box_W}_{\mathcal{H}\otimes\mathbb{1}_W}"] \arrow[d, mapsto, "\underset{W}{\oplus}"'] & \underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[d, mapsto, "\underset{W}{\oplus}"'] & \\
\textbf{effective} & \underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}}"] \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\bigcirc}\mathcal{H} \arrow[d, leftrightarrow, "\wr"'] & \mathcal{H} \arrow[l, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}"'] \arrow[d, leftrightarrow, "\wr"'] \\
\textbf{algebraic} & \mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathcal{H} \arrow[r, "\mathrm{prod}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathcal{H}}", "|w_1,w_2\rangle\otimes|\psi\rangle\mapsto\delta^{w_2}_{w_1}|w_2\rangle\otimes|\psi\rangle"'] & \mathrm{Q}W\otimes\mathcal{H} & \mathbb{1}\otimes\mathcal{H} \arrow[l, "\mathrm{unit}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathcal{H}}"', "\sum_w|w\rangle\otimes|\psi\rangle\mapsfrom 1\otimes|\psi\rangle"]
\end{tikzcd}$$

State preparation

$$\begin{tikzcd}
\textbf{epistemic} & & \underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[r, "\mathrm{ret}^{\Diamond_W}_{\mathcal{H}\otimes\mathbb{1}_W}"] \arrow[d, mapsto, "\underset{W}{\oplus}"'] & \underset{W}{\Diamond}\underset{W}{\mathbb{1}}\otimes\mathcal{H} \arrow[d, mapsto, "\underset{W}{\oplus}"'] \\
\textbf{effective} & \mathcal{H} \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\text{☆}}\mathcal{H}T \arrow[l, "\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}}"'] \arrow[r, "\mathrm{dplc}^{\text{☆}_W}_{\mathcal{H}}"] \arrow[d, leftrightarrow, "\wr"'] & \underset{W}{\text{☆}}\underset{W}{\text{☆}}\mathcal{H} \arrow[d, leftrightarrow, "\wr"'] \\
\textbf{algebraic} & \mathcal{H} & \mathrm{Q}W\otimes\mathcal{H} \arrow[l, "\mathrm{counit}_{\mathrm{Q}W}"', "|\psi\rangle\mapsfrom|w\rangle\otimes|\psi\rangle"] \arrow[r, "\mathrm{coprod}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathcal{H}}", "|w\rangle\otimes|\psi\rangle\mapsto|w\rangle\otimes|w\rangle\otimes|\psi\rangle"'] & \mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathcal{H}
\end{tikzcd}$$

(205)

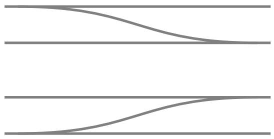
□

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

$$\begin{tikzcd}
\underset{W}{\text{☆}} \arrow[r, "\mathrm{dplc}^{\text{☆}_W}"] \arrow[rr, bend right, "\sim"'] & \underset{W}{\text{☆}}\underset{W}{\text{☆}} \simeq \underset{W}{\bigcirc}\underset{W}{\bigcirc} \arrow[r, "\mathrm{join}^{\bigcirc_W}"] & \underset{W}{\bigcirc}
\end{tikzcd} \tag{206}$$

Remark 2.36 (Frobenius property and Spider theorem)**.** The Frobenius property of $\bigcirc \simeq \text{☆}$ (Prop. 2.35) says explicitly that this diagram commutes:

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



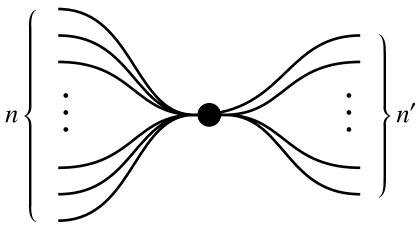
But this already implies (by the theory of *normal forms* [Ab96, Prop. 12, Fig. 3][Ko04], together with specialty (206)) the equality of all those transformations of the form

$$\bigcirc^{n} \longrightarrow \text{☆}^{n'} \tag{207}$$

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 example:

$$\begin{tikzcd}
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\bigcirc_W\mathcal{H}}"] & \underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}}"] & \underset{W}{\bigcirc}\mathcal{H} \simeq \underset{W}{\text{☆}}\mathcal{H} \arrow[r, "\mathrm{dplc}^{\text{☆}_W}_{\mathcal{H}}"] & \underset{W}{\text{☆}}\underset{W}{\text{☆}}\mathcal{H} \\
\mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathcal{H} \arrow[r, "\mathrm{prod}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathrm{Q}W}"] & \mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathcal{H} \arrow[r, "\mathrm{prod}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathcal{H}}"] & \mathrm{Q}W\otimes\mathcal{H} \arrow[r, "\mathrm{coprod}_{\mathrm{Q}W}\otimes\mathrm{id}_{\mathcal{H}}"] & \mathrm{Q}W\otimes\mathrm{Q}W\otimes\mathcal{H}
\end{tikzcd}$$

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 (207) may uniquely by depicted by a diagram of this form:



(208)

These are the *spider diagrams* used in `zxCalculus` (Lit. 1.18).

Indefiniteness as a computational effect. We may now cast these structures into natural programming language constructs for *computational effects* used in §2.4 to encode (quantum gates controlled by) quantum measurement.

Proposition 2.37 (**Indefiniteness modality is strong).**
For W : ClaType *the indefiniteness-modality* $\bigcirc_W$: QuType → QuType *carries symmetric monoidal structure* (78) *as shown in* (210) *exhibiting it as a computational effect* (77)*:*

$$\begin{aligned}
\mathtt{return}^{\bigcirc_W}_{\mathcal{H}} &\;⦂\; \mathcal{H} \multimap \underset{W}{\bigcirc}\mathcal{H} \\
\mathtt{return}^{\bigcirc_W}_{\mathcal{H}} &\;\equiv\; |\psi\rangle \;\mapsto\; \big(w \mapsto |\psi\rangle\big) \\
\mathtt{bind}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'} &\;⦂\; \big(\mathcal{H} \multimap \underset{W}{\bigcirc}\mathcal{H}'\big) \multimap \big(\underset{W}{\bigcirc}\mathcal{H} \multimap \underset{W}{\bigcirc}\mathcal{H}'\big) \\
\mathtt{bind}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'} &\;\equiv\; \Big(|\psi\rangle \mapsto \big(w \mapsto G_w|\psi\rangle\big)\Big) \mapsto \Big(\big(w \mapsto |\psi_w\rangle\big) \mapsto \big(w \mapsto G_w|\psi_w\rangle\big)\Big)
\end{aligned} \tag{209}$$

As such, this monadic effect is the part of the `QS` language in §3 responsible for quantum measurement and classical control.

Dually:

Proposition 2.38 (**Randomness modality is costrong).** *For* W : ClaType *the randomness-modality* $\star_W$: QuType → QuType *carries symmetric comonoidal comonad structure as shown in* (211)*.*

Symmetric monoidal structure on the $\bigcirc_W$-monad (cf. Prop. 2.37):

structure

$$\begin{tikzcd}
(\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}') \arrow[r, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}'}"] & \underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}') \\
(w \mapsto |\psi_w\rangle) \otimes (w' \mapsto |\psi'_{w'}\rangle) \arrow[r, mapsto] & (w \mapsto |\psi_w\rangle \otimes |\psi'_w\rangle)
\end{tikzcd}$$

monad

$$\begin{tikzcd}
\mathcal{H}\otimes\mathcal{H}' \arrow[rrr, "\left(\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}}\right)\otimes\left(\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}'}\right)"] \arrow[ddd, equal] & & & (\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}') \arrow[ddd, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}'}"] \\
& |\psi\rangle \otimes |\psi'\rangle \arrow[r, mapsto] \arrow[d, mapsto] & (w \mapsto |\psi_w\rangle) \otimes (w \mapsto |\psi'_w\rangle) \arrow[d, mapsto] & \\
& |\psi\rangle \otimes |\psi'\rangle \arrow[r, mapsto] & (w \mapsto |\psi_w\rangle \otimes |\psi'_w\rangle) & \\
\mathcal{H}\otimes\mathcal{H}' \arrow[rrr, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}'}"'] & & & \underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}')
\end{tikzcd}$$

$$\begin{tikzcd}
(\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H}') \arrow[rrr, "\left(\mathrm{join}^{\bigcirc_W}_{\mathcal{H}}\right)\otimes\left(\mathrm{join}^{\bigcirc_W}_{\mathcal{H}'}\right)"] \arrow[dd, "\mathrm{pair}^{\bigcirc_W}_{\bigcirc_W\mathcal{H},\,\bigcirc_W\mathcal{H}'}"'] & & & (\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}') \arrow[dddd, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'}"] \\
& ((w,w') \mapsto |\psi_{w,w'}\rangle) \otimes ((w,w') \mapsto |\psi'_{w,w'}\rangle) \arrow[r, mapsto] \arrow[dd, mapsto] & (w \mapsto |\psi_{w,w}\rangle) \otimes (w \mapsto |\psi'_{w,w}\rangle) \arrow[dd, mapsto] & \\
\underset{W}{\bigcirc}\Big((\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}')\Big) \arrow[dd, "\underset{W}{\bigcirc}\left(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'}\right)"'] & & & \\
& ((w,w') \mapsto |\psi_{w,w'}\rangle \otimes |\psi'_{w,w'}\rangle) \arrow[r, mapsto] & (w \mapsto |\psi_{w,w}\rangle \otimes |\psi'_{w,w}\rangle) & \\
\underset{W}{\bigcirc}\underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}') \arrow[rrr, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}'}"'] & & & \underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}')
\end{tikzcd} \tag{210}$$

monoidal

$$\begin{tikzcd}
& & & \left(\underset{W}{\bigcirc}\mathbb{1}\right) \otimes \left(\underset{W}{\bigcirc}\mathcal{H}\right) \arrow[ddd, "\mathrm{pair}^{\bigcirc_W}_{\mathbb{1},\mathcal{H}}"] \\
& & (w \mapsto 1) \otimes (w \mapsto |\psi_w\rangle) \arrow[d, mapsto] & \\
& 1 \otimes (w \mapsto |\psi_w\rangle) \arrow[ur, mapsto] \arrow[r, mapsto] & (w \mapsto |\psi_w\rangle) & \\
\mathbb{1} \otimes \underset{W}{\bigcirc}\mathcal{H} \arrow[rrr, equal] \arrow[uuurrr, bend left, "\mathrm{ret}^{\bigcirc_W}_{\mathbb{1}}\otimes\mathrm{id}"] & & & \underset{W}{\bigcirc}\mathcal{H}
\end{tikzcd}$$

$$\begin{tikzcd}
(\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}') \otimes (\underset{W}{\bigcirc}\mathcal{H}'') \arrow[rrr, "\left(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}'}\right)\otimes\mathrm{id}"] \arrow[ddd, "\mathrm{id}\otimes\left(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'}\right)"'] & & & (\underset{W}{\bigcirc}\mathcal{H}\otimes\mathcal{H}') \otimes (\underset{W}{\bigcirc}\mathcal{H}'') \arrow[ddd, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}\otimes\mathcal{H}',\mathcal{H}''}"] \\
& (w \mapsto |\psi_w\rangle) \otimes (w \mapsto |\psi'_w\rangle) \otimes (w \mapsto |\psi''_w\rangle) \arrow[r, mapsto] \arrow[d, mapsto] & (w \mapsto |\psi_w\rangle \otimes |\psi'_w\rangle) \otimes (w \mapsto |\psi''_w\rangle) \arrow[d, mapsto] & \\
& (w \mapsto |\psi_w\rangle) \otimes (w \mapsto |\psi'_w\rangle \otimes |\psi''_w\rangle) \arrow[r, mapsto] & (w \mapsto |\psi_w\rangle \otimes |\psi'_w\rangle \otimes |\psi''_w\rangle) & \\
(\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}' \otimes \mathcal{H}'') \arrow[rrr, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'\otimes\mathcal{H}''}"'] & & & \underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}'\otimes\mathcal{H}'')
\end{tikzcd}$$

symmetric

$$\begin{tikzcd}
(\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}') \arrow[rrr, "\mathrm{braid}^{\otimes}_{\bigcirc_W\mathcal{H},\bigcirc_W\mathcal{H}'}"] \arrow[ddd, "\mathrm{pair}^{\bigcirc}_{\mathcal{H},\mathcal{H}'}"'] & & & (\underset{W}{\bigcirc}\mathcal{H}') \otimes (\underset{W}{\bigcirc}\mathcal{H}) \arrow[ddd, "\mathrm{pair}^{\bigcirc}_{\mathcal{H}',\mathcal{H}}"] \\
& (w \mapsto |\psi_w\rangle) \otimes (w \mapsto |\psi'_w\rangle) \arrow[r, mapsto] \arrow[d, mapsto] & (w \mapsto |\psi'_w\rangle) \otimes (w \mapsto |\psi_w\rangle) \arrow[d, mapsto] & \\
& (w \mapsto |\psi_w\rangle \otimes |\psi'_w\rangle) \arrow[r, mapsto] & (w \mapsto |\psi'_w\rangle \otimes |\psi_w\rangle) & \\
\underset{W}{\bigcirc}(\mathcal{H}\otimes\mathcal{H}') \arrow[rrr, "\bigcirc_W\left(\mathrm{braid}^{\otimes}_{\mathcal{H},\mathcal{H}'}\right)"'] & & & \underset{W}{\bigcirc}(\mathcal{H}'\otimes\mathcal{H})
\end{tikzcd}$$

Symmetric comonoidal structure on the ☆_W-comonad (cf. Prop. 2.38):

structure

$$\begin{tikzcd}
(\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}') & \underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}') \arrow[l, "\mathrm{copair}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}'}"'] \\
(w,\, |\psi\rangle) \otimes (w,\, |\psi'\rangle) & (w,\, |\psi\rangle \otimes |\psi'\rangle) \arrow[l, mapsto]
\end{tikzcd}$$

comonad

$$\begin{tikzcd}
\mathcal{H} \otimes \mathcal{H}' \arrow[ddd, equal] & & & (\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}') \arrow[lll, "\big(\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}}\big)\otimes\big(\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}'}\big)"'] \\
& |\psi\rangle \otimes |\psi'\rangle & (w,\, |\psi\rangle) \otimes (w|\psi'\rangle) \arrow[l, mapsto] & \\
& |\psi\rangle \otimes |\psi'\rangle \arrow[u, mapsto] & (w,\, |\psi\rangle \otimes |\psi'\rangle) \arrow[l, mapsto] \arrow[u, mapsto] & \\
\mathcal{H} \otimes \mathcal{H}' & & & \underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}') \arrow[lll, "\mathrm{obt}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}'}"] \arrow[uuu, "\mathrm{copair}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}'}"']
\end{tikzcd}$$

$$\begin{tikzcd}
(\underset{W}{\text{☆}}\,\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\,\underset{W}{\text{☆}}\mathcal{H}') & & & (\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}') \arrow[lll, "\big(\mathrm{dupl}^{\text{☆}_W}_{\mathcal{H}}\big)\otimes\big(\mathrm{dupl}^{\text{☆}_W}_{\mathcal{H}'}\big)"'] \\
& ((w,w),\, |\psi\rangle) \otimes ((w,w),\, |\psi'\rangle) & (w,\, |\psi\rangle) \otimes (w,\, |\psi'\rangle) \arrow[l, mapsto] & \\
\underset{W}{\text{☆}}\Big((\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}')\Big) \arrow[uu, "\mathrm{copair}^{\text{☆}_W}_{\text{☆}_W\mathcal{H},\, \text{☆}_W\mathcal{H}'}"] & & & \\
& ((w,w), |\psi\rangle \otimes |\psi'\rangle) \arrow[uu, mapsto] & (w,\, |\psi\rangle \otimes |\psi'\rangle) \arrow[l, mapsto] \arrow[uu, mapsto] & \\
\underset{W}{\text{☆}}\,\underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}') \arrow[uu, "\underset{W}{\text{☆}}\big(\mathrm{copair}^{\text{☆}_W}_{\mathcal{H},\mathcal{H}'}\big)"] & & & \underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}') \arrow[lll, "\mathrm{dupl}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}'}"] \arrow[uuuu, "\mathrm{copair}^{\text{☆}_W}_{\mathcal{H},\mathcal{H}'}"']
\end{tikzcd}$$

(211)

comonoidal

$$\begin{tikzcd}
& & \Big(\underset{W}{\text{☆}}\mathbb{1}\Big) \otimes \Big(\underset{W}{\text{☆}}\mathcal{H}\Big) \arrow[ddd, bend right, "\mathrm{obt}^{\text{☆}_W}_{\mathbb{1}} \otimes \mathrm{id}"'] \\
& (w,\, 1) \otimes (w,\, |\psi\rangle) \arrow[dl, mapsto] & \\
1 \otimes (w,\, |\psi\rangle) & (w,\, |\psi\rangle) \arrow[l, mapsto] \arrow[u, mapsto] & \\
\mathbb{1} \otimes \underset{W}{\text{☆}}\mathcal{H} \arrow[rr, equal] & & \underset{W}{\text{☆}}\mathcal{H} \arrow[uuu, "\mathrm{copair}^{\text{☆}_W}_{\mathbb{1},\mathcal{H}}"']
\end{tikzcd}$$

$$\begin{tikzcd}
(\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}') \otimes (\underset{W}{\text{☆}}\mathcal{H}'') & & & (\underset{W}{\text{☆}}\mathcal{H} \otimes \mathcal{H}') \otimes (\underset{W}{\text{☆}}\mathcal{H}'') \arrow[lll, "\big(\mathrm{copair}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}'}\big)\otimes \mathrm{id}"'] \\
& (w,\, |\psi\rangle) \otimes (w,\, |\psi'_w\rangle) \otimes (w,\, |\psi''\rangle) & (w,\, |\psi\rangle \otimes |\psi'\rangle) \otimes (w,\, |\psi''\rangle) \arrow[l, mapsto] & \\
& (w,\, |\psi\rangle) \otimes (w,\, |\psi'\rangle \otimes |\psi''\rangle) \arrow[u, mapsto] & (w,\, |\psi\rangle \otimes |\psi'\rangle \otimes |\psi''\rangle) \arrow[l, mapsto] \arrow[u, mapsto] & \\
(\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}' \otimes \mathcal{H}'') \arrow[uuu, "\mathrm{id}\otimes\big(\mathrm{copair}^{\text{☆}_W}_{\mathcal{H},\mathcal{H}'}\big)"] & & & \underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}' \otimes \mathcal{H}'') \arrow[lll, "\mathrm{copair}^{\bigcirc_W}_{\mathcal{H},\mathcal{H}'\otimes\mathcal{H}''}"] \arrow[uuu, "\mathrm{copair}^{\text{☆}_W}_{\mathcal{H}\otimes\mathcal{H}',\mathcal{H}''}"']
\end{tikzcd}$$

symmetric

$$\begin{tikzcd}
(\underset{W}{\text{☆}}\mathcal{H}) \otimes (\underset{W}{\text{☆}}\mathcal{H}') & & & (\underset{W}{\text{☆}}\mathcal{H}') \otimes (\underset{W}{\text{☆}}\mathcal{H}) \arrow[lll, "\mathrm{braid}^{\otimes}_{\text{☆}_W\mathcal{H},\, \text{☆}_W\mathcal{H}'}"'] \\
& (w,\, |\psi\rangle) \otimes (w,\, |\psi'\rangle) & (w,\, |\psi'\rangle) \otimes (w,\, |\psi\rangle) \arrow[l, mapsto] & \\
& (w,\, |\psi\rangle \otimes |\psi'\rangle) \arrow[u, mapsto] & (w,\, |\psi'\rangle \otimes |\psi\rangle) \arrow[l, mapsto] \arrow[u, mapsto] & \\
\underset{W}{\text{☆}}(\mathcal{H} \otimes \mathcal{H}') \arrow[uuu, "\mathrm{copair}^{\text{☆}}_{\mathcal{H},\mathcal{H}'}"] & & & \underset{W}{\text{☆}}(\mathcal{H}' \otimes \mathcal{H}) \arrow[lll, "\text{☆}_W\big(\mathrm{braid}^{\otimes}_{\mathcal{H},\mathcal{H}'}\big)"] \arrow[uuu, "\mathrm{copair}^{\text{☆}}_{\mathcal{H}',\mathcal{H}}"']
\end{tikzcd}$$

In outlook to the discussion of mixed quantum states in §2.5 we close this section on quantum epistemology by observing that indefiniteness- and randomness-effects lift from pure to mixed quantum states via the above (co)monoidal (co)monad structure, via the monoidal monad structure $\mathrm{pair}^{\bigcirc_W}$ (210) on the indefinite modality and the comonoidal comonad structure $\mathrm{copair}^{\text{☆}_W}$ (211) on the random modality.

Indefinite mixed states. A quantum system with pure state space $\mathcal{H} \,:\, \mathrm{QuType}^{\mathrm{fdm}}$ a dualizable (133) quantum type generally has *mixed* states (35) in $\mathcal{H} \otimes \mathcal{H}^* \,:\, \mathrm{QuType}$, such that a quantum gate on pure states induces a *quantum channel* on mixed states, of the form

$$A \,:\, \mathcal{H}_1 \to \mathcal{H}_2 \qquad \vdash \qquad \mathrm{chan}^A \,:\, \begin{array}{c}\mathcal{H}_1\\ \otimes\\ \mathcal{H}_1^*\end{array} \xrightarrow{\begin{array}{c} f \\ \otimes \\ f^{\dagger *}\end{array}} \begin{array}{c}\mathcal{H}_2\\ \otimes\\ \mathcal{H}_2^*\end{array}, \tag{212}$$

(for the moment the dagger-$(-)^\dagger$ operation may be treated as a black box, we discuss this in [SS23-QR]).

Lemma 2.39 (**Enhancing indefiniteness-effects to Mixed states).** *The assignment which sends an $\bigcirc_W$-effectful map to its tensor product with its adjoint dual* (212) *followed by the $\bigcirc_W$-pairing* (210)

$$\begin{tikzcd}
\mathcal{H}_1 \arrow[rr, "G_\bullet"] & {} \arrow[d, mapsto] & \underset{W}{\bigcirc}\mathcal{H}_2 \\
\begin{array}{c}\mathcal{H}_1\\ \otimes\\ \mathcal{H}_1^*\end{array} \arrow[r, "\begin{array}{c}G_\bullet\\ \otimes\\ G_\bullet^{\dagger *}\end{array}"] & \begin{array}{c}\underset{W}{\bigcirc}\mathcal{H}_2\\ \otimes\\ \underset{W}{\bigcirc}\mathcal{H}_2^*\end{array} \arrow[r, "\mathrm{pair}^{\bigcirc_W}"] & \underset{W}{\bigcirc}\begin{array}{c}\mathcal{H}_2\\ \otimes\\ \mathcal{H}_2^*\end{array}
\end{tikzcd} \tag{213}$$

preserves $\bigcirc_W$-Kleisli-composition (69)*, in that:*

$$\big(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_2,\, \mathcal{H}_2^*} \circ (G_\bullet \otimes G_\bullet^{\dagger *})\big) >=> \big(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\, \mathcal{H}_3^*} \circ (H_\bullet \otimes H_\bullet^{\dagger *})\big) \quad = \quad \mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\, \mathcal{H}_3^*} \circ \Big((G_\bullet >=> H_\bullet) \otimes (G_\bullet^{\dagger *} >=> H_\bullet^{\dagger *})\Big) \tag{214}$$

and hence defines a faithful endofunctor on the free $\bigcirc_W$-modales (93)

$$\mathrm{pair}^{\bigcirc_W} \circ (-)_\bullet \otimes (-)_\bullet^{\dagger *} \;:\; \mathrm{QuType}_{\bigcirc_W} \longrightarrow \mathrm{QuType}_{\bigcirc_W} \tag{215}$$

Proof. This is an argument analogous to that for monad transformations (105). Consider the following diagram:

$$\begin{tikzcd}
\mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "G_\bullet \otimes G_\bullet^{\dagger *}"] & (\underset{W}{\bigcirc}\mathcal{H}_2) \otimes (\underset{W}{\bigcirc}\mathcal{H}_2^*) \arrow[r, "(\bigcirc_W H_\bullet) \otimes (\bigcirc_W H_\bullet^{\dagger *})"] \arrow[d, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}, \mathcal{H}^*}"] & (\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H}_3) \otimes (\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{H}_3^*) \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}} \otimes \mathrm{join}^{\bigcirc_W}_{\mathcal{H}^*}"] \arrow[d, "\mathrm{pair}^{\bigcirc_W}_{\bigcirc_W \mathcal{H}_3,\, \bigcirc_W \mathcal{H}_3^*}"] & (\underset{W}{\bigcirc}\mathcal{H}) \otimes (\underset{W}{\bigcirc}\mathcal{H}^*) \arrow[dd, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\, \mathcal{H}_3^*}"] \\
& \mathcal{H}_2 \otimes \mathcal{H}_2^* \arrow[r, "\bigcirc_W\left(H_\bullet \otimes H_\bullet^{\dagger *}\right)"] & \underset{W}{\bigcirc}\Big((\underset{W}{\bigcirc}\mathcal{H}_3) \otimes (\underset{W}{\bigcirc}\mathcal{H}_3^*)\Big) \arrow[d, "\underset{W}{\bigcirc}\left(\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_3,\, \mathcal{H}_3^*}\right)"] & \\
& & \underset{W}{\bigcirc}\underset{W}{\bigcirc}(\mathcal{H}_3 \otimes \mathcal{H}_3^*) \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathcal{H}_3,\, \mathcal{H}_3^*}"] & \underset{W}{\bigcirc}(\mathcal{H}_3 \otimes \mathcal{H}_3^*)
\end{tikzcd} \tag{216}$$

Here the middle square commutes by the naturality of the pairing map, while the right square commutes as part of the monoidal monad structure (210) exhibited by the pairing. Therefore the full diagram commutes. Since its total top and right composite is the right hand side of (214) while its total left and bottom (diagonal) composite is the left hand side of (214), this proves the claim. ◻

2.4 Quantum Gates & Measurement

We explain how *controlled quantum gates* and *quantum measurement gates* (Lit. 1.1) are naturally represented in the quantum modal logic of §2.3 and give (Prop. 2.40) a formal proof of the *deferred measurement principle* (18).

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

We may observe that, with §2.3, 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 $\bigcirc$ ($\Box$).

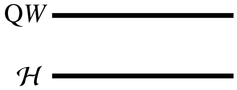
Throughout we use monadicity of $\oplus_W$ (Prop. 2.32) to translate (204)

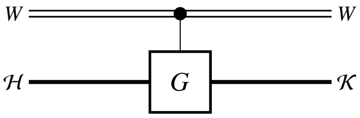
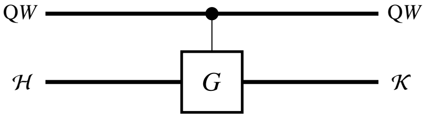
- *epistemic typing*
  via W-dependent linear types

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 $\dfrac{\underset{W}{\Box}\mathcal{H}_\bullet \;:\; \mathrm{QuType}_W}{w : W \;\vdash\; \underset{w'}{\oplus}\mathcal{H}_{w'} \;:\; \mathrm{QuType}}$
Effective	indefiniteness-handling quantum data $\begin{tikzcd} \underset{W}{\oplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] & : & \mathrm{QuType}^{\bigcirc_W} \end{tikzcd}$	free indefiniteness-handling quantum data $\begin{tikzcd} \underset{W}{\oplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] & : & \mathrm{QuType}^{\bigcirc_W} \\ \underset{W}{\bigcirc}\underset{W}{\oplus}\mathcal{H}_\bullet & : & \mathrm{QuType}_{\bigcirc_W} \arrow[u, hook] \end{tikzcd}$

	Classically controlled quantum gate	**Quantumly controlled quantum gate**
Symbolic		
Epistemic	$\begin{tikzcd} & \mathcal{H}_\bullet \arrow[r, "G_\bullet", "\text{an actual entailment}"'] & \mathcal{K}_\bullet \\ w : W \;\vdash & \mathcal{H}_w \arrow[r, "G_w"] & \mathcal{K}_w \end{tikzcd}$	$\begin{tikzcd} & \underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\Box}G_\bullet", "\text{a potential entailment}"'] & \underset{W}{\Box}\mathcal{K}_\bullet \\ w : W \;\vdash & \underset{W}{\oplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\oplus}G_\bullet" description] & \underset{W}{\oplus}\mathcal{K}_\bullet \end{tikzcd}$
Effective	$\begin{tikzcd} \underset{W}{\oplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\oplus}G_\bullet"] \arrow[d, equal, "\text{if } \mathcal{H}_\bullet = \mathcal{H}"'] & \underset{W}{\oplus}\mathcal{K}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal, "\text{if } \mathcal{K}_\bullet = \mathcal{K}"'] \\ \underset{W}{\bigcirc}\mathcal{H} \arrow[r, "\underset{W}{\bigcirc}G_\bullet", "{\mathrm{bind}\big(\mathcal{H} \xrightarrow{G_\bullet} \bigcirc_W \mathcal{K}\big)}"'] & \underset{W}{\bigcirc}\mathcal{K} \end{tikzcd}$ a $\bigcirc$-effectful operation	$\begin{tikzcd} \underset{W}{\oplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\oplus}\underset{W}{\Box}G_\bullet"] \arrow[d, equal] & \underset{W}{\oplus}\underset{W}{\Box}\mathcal{K}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] \\ \underset{W}{\bigcirc}\underset{W}{\oplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\bigcirc}\underset{W}{\oplus}G_\bullet", "{\mathrm{bind}\big(\mathrm{return} \circ \underset{W}{\oplus}G_\bullet\big)}"'] & \underset{W}{\bigcirc}\underset{W}{\oplus}\mathcal{K}_\bullet \end{tikzcd}$ 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 (204) "effective" typing in the top line of the bottom row follows by monadicity of $\oplus_W$ (see Prop. 2.32). The very last line shows the corresponding Kleisli-triple formulation of "programs with side effects" (67). 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. 81) since that makes it correspond to a free $\bigcirc$-modale. On the right we see the effectless operation (70).

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. 2.29), as shown on the right.

Via the language of effectful computation (Lit. 1.17) and with the "reader-monad" $\bigcirc$ modally pronounced as "indefiniteness" (195), this translates to the pleasant statement that:

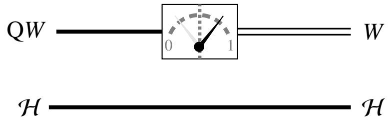
"For effectively-typed quantum data, quantum measurement is nothing but the *handling of indefiniteness-effects*" (regarded as modale homorphisms via (94)).

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" (21) is resolved into an appropriate quantum effect language equivalent (204) to quantum modal logic.

Quantum measurement gate

Symbolic



Epistemic

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

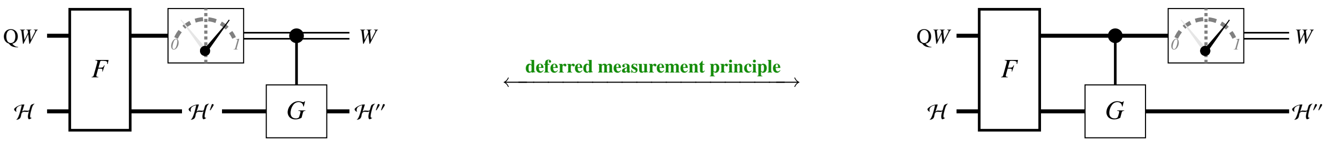
Effective

$$\begin{tikzcd} \underset{W}{\bigoplus}\underset{W}{\Box}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[r, "\underset{W}{\oplus}\mathrm{obt}^{\underset{W}{\Box}}_{\mathcal{H}_\bullet}"] \arrow[d, equal] \arrow[dr, phantom, "(97)"] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[loop above, "\bigcirc_W"] \arrow[d, equal] \\ \underset{W}{\bigcirc}\underset{W}{\bigoplus}\mathcal{H}_\bullet \arrow[r, "\mathtt{handle}^{\bigcirc_W}_{\oplus_W \mathcal{H}_\bullet}", "\bigcirc_W\text{-effect handling}"'] & \underset{W}{\bigoplus}\mathcal{H}_\bullet \end{tikzcd}$$

Before looking at examples (p. 81), we record a basic structural result immediately implied by this typing, which may evidently be understood as formalizing the *deferred measurement principle* (18), thus making this principle verifiable in `LHoTT` as [Sta15] envisioned should be the case for any respectable quantum programming language:

Proposition 2.40 (Deferred measurement principle). *With respect to the above typing of quantum gates, the* $\Box$*-Kleisli equivalence* (95) *is the following transformation of quantum circuits:*

$$\begin{tikzcd}[column sep=large] (\mathrm{QuType}_W)_{\Box_W}(\mathcal{H}_\bullet, \mathcal{H}''_\bullet) \arrow[r, "\sim", "\underset{W}{\Box}(\text{-}) \circ \mathrm{dplc}^{\Box_W}_{(\text{-})}"'] \arrow[rr, bend left, "\mathrm{id}", "\text{Kleisli equivalence}"'] & \mathrm{QuType}^{\Box_W}_W(\underset{W}{\Box}\mathcal{H}_\bullet, \underset{W}{\Box}\mathcal{H}''_\bullet) \arrow[r, "\sim", "\mathrm{obt}^{\Box_W}_{(\text{-})} \circ (-)"'] & (\mathrm{QuType}_W)_{\Box_W}(\mathcal{H}_\bullet, \mathcal{H}''_\bullet) \\ \underset{W}{\Box}\text{-Kleisli morphisms} & \text{homomorphisms of free } \underset{W}{\Box}\text{-coalgebras} & \underset{W}{\Box}\text{-Kleisli morphisms} \\ \left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\mathrm{obt}^{\Box_W}_{\mathcal{H}'_\bullet}} \mathcal{H}'_\bullet \xrightarrow{G_\bullet} \mathcal{H}''_\bullet\right) \arrow[r, mapsto] & \left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\Box_W G_\bullet} \underset{W}{\Box}\mathcal{H}''_\bullet\right) \arrow[r, mapsto] & \left(\underset{W}{\Box}\mathcal{H}_\bullet \xrightarrow{F} \underset{W}{\Box}\mathcal{H}'_\bullet \xrightarrow{\Box_W G_\bullet} \underset{W}{\Box}\mathcal{H}''_\bullet \xrightarrow{\mathrm{obt}^{\Box_W}_{\mathcal{H}''_\bullet}} \mathcal{H}''_\bullet\right) \\ \text{measurement-controlled quantum gate} & \text{quantum-controlled quantum gate...} & \text{...followed by measurement} \end{tikzcd}$$



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:

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

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. 2.31), while the triangle commutes by the comonad axioms (72). $\Box$

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 (17). 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:

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

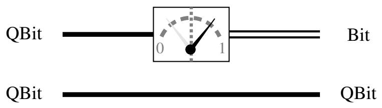
In seeing how the modal typing shown on the right and below matches the standard formulas (17) we repeatedly make use of the following canonical identifications:

$$\begin{aligned} & \mathrm{QBit} \otimes \mathrm{QBit} \\ \simeq\; & \mathbb{C}[\mathrm{Bit}] \otimes \mathrm{QBit} \\ \simeq\; & (\mathbb{C}_{0} \oplus \mathbb{C}_{1}) \otimes \mathrm{QBit} \\ \simeq\; & \mathrm{QBit}_{0} \oplus \mathrm{QBit}_{1} \\ \simeq\; & \textstyle\bigoplus_{\mathrm{Bit}} \mathrm{QBit}_{\bullet} \\ \simeq\; & \bigcirc_{\mathrm{Bit}} \mathrm{QBit}\;, \end{aligned}$$

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

QBit-Measurement

symbolic



epistemic

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

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

Effective

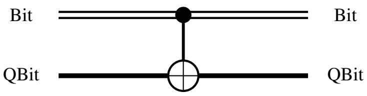
$$\begin{tikzcd} \underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \arrow[rr, "\mathtt{handle}^{\bigcirc_{\mathrm{Bit}}}_{\bigcirc_{\mathrm{Bit}}\mathrm{QBit}}"] & & \underset{\mathrm{Bit}}{\bigcirc}\,\mathrm{QBit} \end{tikzcd}$$

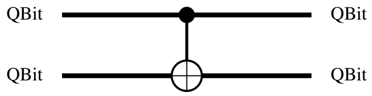
$$\begin{tikzcd} \mathrm{QBit}{\otimes}\mathrm{QBit} \arrow[dr, "P_{0}\otimes\mathrm{id}"] & \\ \oplus & \mathrm{QBit}{\otimes}\mathrm{QBit} \\ \mathrm{QBit}{\otimes}\mathrm{QBit} \arrow[ur, "P_{1}\otimes\mathrm{id}"'] & \end{tikzcd}$$

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

CNOT gate

Symbolic





Epistemic

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

$$b : \mathrm{Bit} \;\vdash\; \begin{tikzcd} \mathrm{QBit} \arrow[rr] & & \mathrm{QBit} \\ \vert b_2\rangle \arrow[rr, mapsto] & & \vert b \;\mathrm{xor}\; b_2\rangle \end{tikzcd}$$

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

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

Effective

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

$$\begin{tikzcd} \mathrm{QBit} \arrow[rr, "\vert b_2\rangle \mapsto \vert 0 \;\mathrm{xor}\; b_2\rangle"] & & \mathrm{QBit} \\ \oplus & \oplus & \oplus \\ \mathrm{QBit} \arrow[rr, "\vert b_2\rangle \mapsto \vert 1 \;\mathrm{xor}\; b_2\rangle"'] & & \mathrm{QBit} \end{tikzcd}$$

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

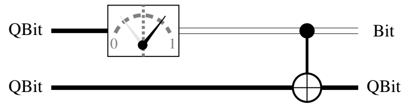
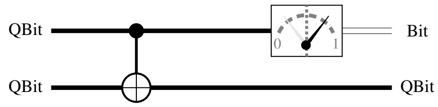
$$\begin{tikzcd} \underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\oplus}\,\mathrm{QBit} \arrow[rr, "\underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\oplus}\,\mathrm{CNOT}_{\bullet}"] & & \underset{\mathrm{Bit}}{\bigcirc}\,\underset{\mathrm{Bit}}{\oplus}\,\mathrm{QBit} \end{tikzcd}$$

$$\begin{tikzcd} \mathrm{QBit}{\otimes}\mathrm{QBit} \arrow[rr, "\vert b_1, b_2\rangle \mapsto \vert b_1, b_1 \;\mathrm{xor}\; b_2\rangle"] & & \mathrm{QBit}{\otimes}\mathrm{QBit} \\ \oplus & \oplus & \oplus \\ \mathrm{QBit}{\otimes}\mathrm{QBit} \arrow[rr, "\vert b_1, b_2\rangle \mapsto \vert b_1, b_1 \;\mathrm{xor}\; b_2\rangle"'] & & \mathrm{QBit}{\otimes}\mathrm{QBit} \end{tikzcd}$$

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

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

CNOT with QBit-Measurement

symbolic		
epistemic	$\underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_\bullet \xrightarrow[]{\overset{\textbf{measurement}}{\mathrm{obt}^{\Box_W}_{\mathrm{QBit}_\bullet}}} \mathrm{QBit}_\bullet \xrightarrow{\overset{\textbf{cls. contr. qnt. NOT}}{\mathrm{CNOT}_\bullet}} \mathrm{QBit}_\bullet$ $b : \mathrm{Bit} \;\;\vdash\;\; \mathrm{QBit}\otimes\mathrm{QBit} \longrightarrow \mathrm{QBit}_b \longrightarrow \mathrm{QBit}_b$ $\lvert b_1\rangle\otimes\lvert b_2\rangle \;\mapsto\; \lvert b_2\rangle \;\mapsto\; \lvert b \,\mathrm{xor}\, b_2\rangle$	$\underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_\bullet \xrightarrow{\overset{\textbf{quantum CNOT}}{\underset{\mathrm{Bit}}{\Box}\mathrm{CNOT}_\bullet}} \underset{\mathrm{Bit}}{\Box}\mathrm{QBit}_\bullet \xrightarrow{\overset{\textbf{measurement}}{\mathrm{obt}^{\Box_{\mathrm{Bit}}}_{\mathrm{QBit}_\bullet}}} \mathrm{QBit}_\bullet$ $b : \mathrm{Bit} \;\;\vdash\;\; \mathrm{QBit}\otimes\mathrm{QBit} \longrightarrow \mathrm{QBit}\otimes\mathrm{QBit} \longrightarrow \mathrm{QBit}_b$ $\lvert b_1\rangle\otimes\lvert b_2\rangle \;\mapsto\; \lvert b_1\rangle\otimes\lvert b_2 \,\mathrm{xor}\, b_1\rangle \;\mapsto\; \lvert b_2 \,\mathrm{xor}\, b\rangle$
Effective	$\underset{\mathrm{Bit}}{\bigcirc}\underset{\mathrm{Bit}}{\oplus}\mathrm{QBit} \xrightarrow{\mathtt{handle}^{\bigcirc_{\mathrm{Bit}}}_{\mathrm{QBit}}} \underset{\mathrm{Bit}}{\oplus}\mathrm{QBit} \xrightarrow{\underset{\mathrm{Bit}}{\oplus}\mathrm{CNOT}_\bullet} \underset{\mathrm{Bit}}{\oplus}\mathrm{QBit}$ $\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[dr, "P_0\otimes\mathrm{id}"] & & \\ \oplus & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \lvert b_1\rangle\otimes\lvert b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "P_1\otimes\mathrm{id}"'] & & \end{tikzcd}$ $\begin{tikzcd} & & \mathrm{QBit} \arrow[r, "\lvert b_2\rangle \mapsto \lvert 0 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit} & \\ \mathrm{bind}\Big(& \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "\lvert b_1, b_2\rangle \mapsto \delta_0^{b_1}\,\lvert b_2\rangle"] \arrow[dr, "\lvert b_1, b_2\rangle \mapsto \delta_1^{b_1}\,\lvert b_2\rangle"'] & \oplus & \oplus & \Big) \\ & & \mathrm{QBit} \arrow[r, "\lvert b_2\rangle \mapsto \lvert 1 \,\mathrm{xor}\, b_2\rangle"'] & \mathrm{QBit} & \end{tikzcd}$	$\underset{\mathrm{Bit}}{\bigcirc}\underset{\mathrm{Bit}}{\oplus}\mathrm{QBit} \xrightarrow{\underset{\mathrm{Bit}}{\bigcirc}\underset{\mathrm{Bit}}{\oplus}\mathrm{QBit}} \underset{\mathrm{Bit}}{\bigcirc}\underset{\mathrm{Bit}}{\oplus}\mathrm{QBit} \xrightarrow{\mathtt{handle}^{\bigcirc}_{\oplus_{\mathrm{Bit}}\mathrm{QBit}}} \underset{\mathrm{Bit}}{\oplus}\mathrm{QBit}$ $\begin{tikzcd} \mathrm{QBit}\otimes\mathrm{QBit} \arrow[rr, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \lvert b_1\rangle\otimes\lvert b_1 \,\mathrm{xor}\, b_2\rangle"] & & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[dr, "P_0\otimes\mathrm{id}"] & \\ \oplus & \oplus & \oplus & \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{QBit}\otimes\mathrm{QBit} \arrow[rr, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \lvert b_1\rangle\otimes\lvert b_1 \,\mathrm{xor}\, b_2\rangle"'] & & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "P_1\otimes\mathrm{id}"'] & \end{tikzcd}$ $\begin{tikzcd} & & & \mathrm{QBit} & \\ \mathrm{bind}\Big(& \mathrm{QBit}\otimes\mathrm{QBit} \arrow[r, "\lvert b_1\rangle\otimes\lvert b_2\rangle \mapsto \lvert b_1\rangle\otimes\lvert b_1 \,\mathrm{xor}\, b_2\rangle"] & \mathrm{QBit}\otimes\mathrm{QBit} \arrow[ur, "\lvert b_1, b_2\rangle \mapsto \delta_0^{b_1}\,\lvert b_2\rangle"] \arrow[dr, "\lvert b_1, b_2\rangle \mapsto \delta_0^{b_1}\,\lvert b_2\rangle"'] & \oplus & \Big) \\ & & & \mathrm{QBit} & \end{tikzcd}$

Notice here how the component expressions on the left and right agree, in accord with the *deferred measurement principle* (Prop. 2.40). 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* (20). Namely if, with (86), we recall that RAM-effects are typed by the state monad $\underset{W}{\bigcirc}\underset{W}{\star}$ (196) — which immediately makes sense linearly just as it does classically—, then quantumly controlled quantum circuits in the above sense (p. 80) are formally identified with QRAM-effective quantum programs as follows, where the first transformation is for effectless programs (70) while the second is $\star_W \dashv \bigcirc_W$-adjointness (75):

The passage to circuit models for qRAM (20) 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 $\underset{W}{\oplus}\mathcal{H} \longmapsto \underset{W}{\oplus}\mathcal{K}$

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

QW-controlled quantum gate (p. 80)

$$\begin{tikzcd} \underset{W}{\oplus}\mathcal{H} \arrow[r, "\underset{W}{\oplus}G_\bullet"] \arrow[d, equal] & \underset{W}{\oplus}\mathcal{K} \arrow[d, equal] \\ \underset{W}{\star}\mathcal{H} \arrow[r, "\underset{W}{\oplus}G_\bullet"] & \underset{W}{\star}\mathcal{K} \end{tikzcd} \tag{217}$$

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

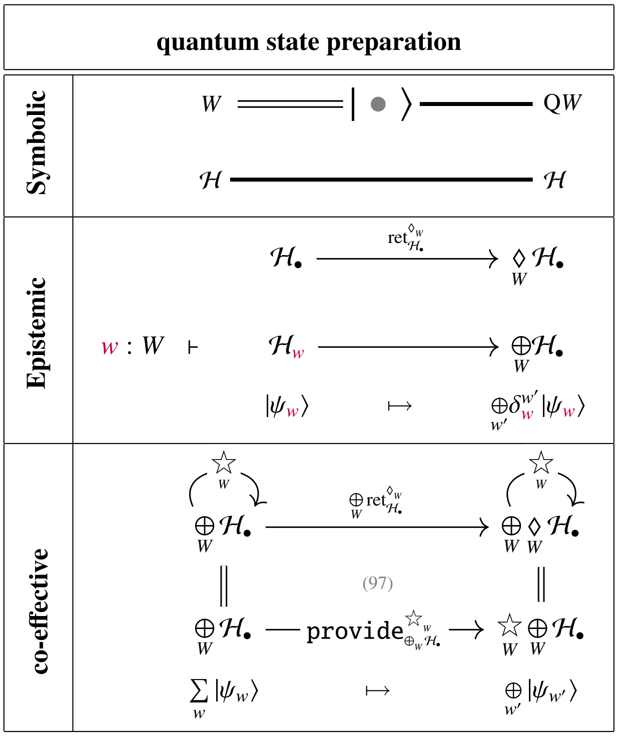
quantum circuit interacting with a QRAM space QW

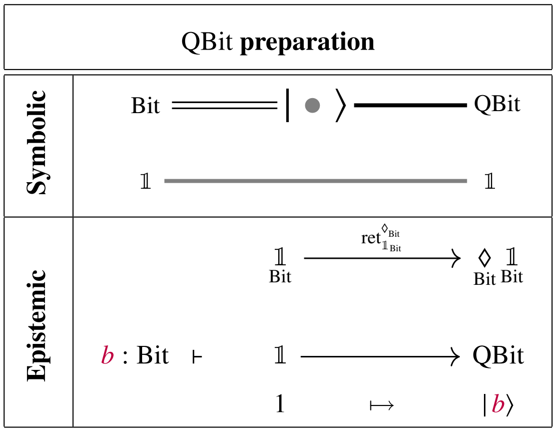
At the same time, this QuantumState-monad

$$QW\mathrm{State} \;\;\simeq\;\; \underset{W}{\bigcirc}\underset{W}{\star}$$

reflects mixed QW-states, discussed in §2.5.

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* (116): 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:





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 $(\text{-}) \otimes \mathbb{1}_W$. 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. 81) by the *hexagon of epistemic entailments* (2.3):

$$\begin{tikzcd}
 & \underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\Box}G_\bullet"] & \underset{W}{\Box}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\Box}\mathrm{ret}^{\lozenge_W}_{\mathcal{H}_\bullet}", "\substack{\text{quantum measurement}\\ \text{typed Everett-style}}"'] & \underset{W}{\Box}\underset{W}{\lozenge}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\Box}\underset{W}{\lozenge}G_\bullet", "1"'] & \underset{W}{\Box}\underset{W}{\lozenge}\mathcal{H}_\bullet \\
w : W \;\vdash & \underset{W}{\oplus}\mathcal{H}_\bullet \arrow[r, "\underset{W}{\oplus}G_\bullet", "\substack{\text{coherent quantum}\\ \text{gate}}"'] \arrow[ddd, phantom, "\equiv" description] & \underset{W}{\oplus}\mathcal{H}_\bullet \arrow[r, "\mathtt{provide}^{\star_W}_{\oplus_W \mathcal{H}_\bullet}", "\substack{\text{collapsed-state-preparation}\\ \text{by providing } \star\text{-context}}"'] \arrow[ddd, phantom, "\equiv" description] & \underset{W}{\star}\underset{W}{\oplus}\mathcal{H}_\bullet \arrow[loop above, "\star_W"] \arrow[r, "\underset{W}{\star}\underset{W}{\oplus}G_\bullet", "\substack{\text{coherent quantum gate}\\ \text{under } \star}"'] \arrow[d, phantom, "\equiv" description] & \underset{W}{\star}\underset{W}{\oplus}\mathcal{H}_\bullet \arrow[loop above, "\star_W"] \arrow[d, phantom, "\equiv" description] \\
 & & & \mathcal{H} \arrow[r, "G"] & \mathcal{H} \\
 & & & \oplus & \oplus \\
 & \mathcal{H} \arrow[r, "G"] & \mathcal{H} \arrow[ruuu, "P_1"] \arrow[rddd, "P_{|W|}"'] \arrow[r, phantom, "\vdots\ \text{branching}"] & \vdots & \vdots \\
 & & & \oplus & \oplus \\
 & & & \mathcal{H} \arrow[r, "G"] & \mathcal{H} \\
 & & \sum_{w'} |\psi_{w'}\rangle \arrow[r, mapsto] & \underset{w''}{\oplus}|\psi_{w''}\rangle \arrow[r, mapsto] & \underset{w''}{\oplus} G_{w''}|\psi_{w''}\rangle
\end{tikzcd}$$

(218)

Remark 2.41 (**No classical control appears in Everett-typing).**
(i) Comparing the epistemic hexagon (7), we find that where the Copenhagen-style typing sees a classically-controlled quantum gate (cf. p. 80) the Everett-style typing (218) sees (no classical control) but the corresponding quantumly-controlled quantum gate — but applied in each of several "branches".
(ii) This primacy of the non-classical quantum perspective and the disregard for the need for 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 (218) provides a formalization of this intuition in a precise and machine-verifiable form.

Remark 2.42 (**Everett-style measurement typing in the literature).**
(i) Essentially the typing-by-branching of quantum measurement in the bottom of (218) may be recognized in the early proposal for quantum programming language syntax in [Se04, p. 568].
(ii) The observation (apparently independently of [Se04]) that this may usefully be understood as the `provide`-operation of modales (coalgebras) over the comonad $\text{☆}_W \simeq QW \otimes (\text{-})$ (Prop. 2.35) 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. 2.36).
(iii) While — in formulating the quantum language `QS` below in §3 — we focus on language constructs for the Copenhagen-style typing (since this brings out the desired *dynamic lifting* of quantum-to-classical control, Lit. 1.11), the situation (218) 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`.

Computational quantum measurement as entering the Indefiniteness-monad. In summary, we have seen that coherent quantum gates are naturally typed as *free* indefinite-effectful linear maps, with quantum measurement given by the handling of the free indefiniteness-effect. Computationally this means equivalently that coherent quantum gates are equivalently the plain linear maps that one expects them to be, with quantum measurement being the step of "entering the indefiniteness"-monad, in the sense of the commutativity of the following diagram:

Computationally, the $\bigcirc$-effective typing of quantum gates with quantum measurement amounts to regarding the map

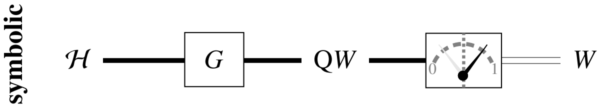
$$\begin{array}{rcl} \texttt{collapse}_W : QW \otimes \mathcal{H} & \longrightarrow & \underset{W}{\bigcirc}\mathcal{H} \\ |w\rangle \otimes |\psi\rangle & \mapsto & (w, |\psi\rangle) \end{array} \tag{219}$$

(whose underlying function is the identity, up to re-typing) as passing into (the category of free modales over) the $\bigcirc_W$-monad, as shown by the commuting diagram on the right.
It is this final computational typing of quantum measurement which neatly lends itself to programming language-articulation in §3, see p. 104.

Notice that while the epistemic, effective and computational perspectives are all equivalent, they superficially express a different ontology of the measurement collapse:

In the epistemic and effective perspective the *eventual* measurement in the W-basis is declared (possibly long) before that measurement takes places: In this perspective all possible future measurement outcomes are pre-emptively allocated in classical data.

Quantum Gate followed by Measurement



$$\begin{tikzcd}
\textbf{epistemic} & QV \otimes \underset{W}{\mathbb{1}} \arrow[r, "G \otimes \underset{W}{\mathbb{1}}"] \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] & \underset{W}{\Box}\underset{W}{\mathbb{1}} \arrow[r, "\mathrm{obt}^{\Box_W}_{\mathbb{1}_W}"] \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] & \underset{W}{\mathbb{1}} \arrow[d, mapsto, "\underset{W}{\bigoplus}"'] \\
\textbf{effective} & \underset{W}{\bigcirc}QV \arrow[r, "\underset{W}{\bigcirc}G"] & \underset{W}{\bigcirc}QW \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathbb{1}}", "\texttt{measure}"'] & \underset{W}{\bigcirc}\mathbb{1} \\
\textbf{computational} & QV \arrow[u, "\mathrm{ret}^{\bigcirc_W}_{QV}"', "\textbf{dynamic lifting}"] \arrow[r, "G", "\textbf{coherent quantum gate}"] & QW \arrow[u, "\mathrm{ret}^{\bigcirc_W}_{QW}", "\texttt{definitely}"'] \arrow[ur, bend right, "\texttt{collapse}"'] &
\end{tikzcd}$$

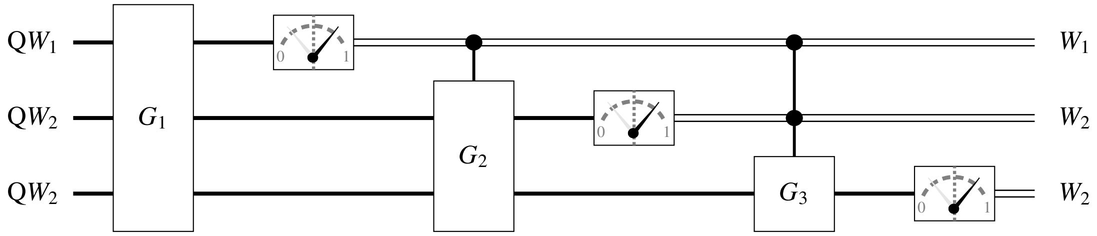
In contrast, in the computational typing the "dynamically lifted" classical measurement outcomes are syntactically referenced only the moment that the measurement actually takes place (computationally). In particular, as successive quantum measurements are made, the computational typing of the quantum circuit accumulates the corresponding indefiniteness-modalities, reflecting the fact that more and more measurement outcomes $w_i : W_i$ become "dynamically lifted" into the classical register (Lit. 1.11):

Computational typing of successive dynamically lifted quantum measurementss

$$\mathrm{Q}W_1 \otimes \mathrm{Q}W_2 \otimes \mathrm{Q}W_3 \xrightarrow{G_1;\mathtt{collapse}_{W_1}} \underset{W_1}{\bigcirc}\mathrm{Q}W_2 \otimes \mathrm{Q}W_3 \xrightarrow{G_2;\mathtt{collapse}_{W_2}} \underset{W_1}{\bigcirc}\,\underset{W_2}{\bigcirc}\,\mathrm{Q}W_3 \xrightarrow{G_3;\mathtt{collapse}_{W_3}} \underset{W_1}{\bigcirc}\,\underset{W_2}{\bigcirc}\,\underset{W_3}{\bigcirc}\,\mathbb{1}$$

data in quantum registers

data in classical registers



(220)

Enhancing dynamically lifted quantum measurement from pure to mixed states. Remarkably, the above effective and computational typing of quantum measurement and controlled quantum gates is enhanced *verbatim* to quantum channels on mixed states (35), due to the faithful functor (215)

$$\begin{tikzcd}
\mathrm{pair}^{\bigcirc_W} \circ (-) \otimes (-)^{\dagger *} \;:\; \mathrm{QuType}_{\bigcirc_W} \arrow[r] & \mathrm{QuType}_{\bigcirc_W} \\
\mathcal{H}_1 \arrow[dd, "A_\bullet"'] & \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[d, "A_\bullet \otimes A_\bullet^{\dagger *}"] \\
 & \big(\underset{W}{\bigcirc}\mathcal{H}_2\big) \otimes \big(\underset{W}{\bigcirc}\mathcal{H}_2^*\big) \arrow[d, "\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_2,\mathcal{H}_2^*}"] \\
\underset{W}{\bigcirc}\mathcal{H}_2 & \underset{W}{\bigcirc}(\mathcal{H}_2 \otimes \mathcal{H}_2^*)\,.
\end{tikzcd}$$

In the same manner, the computational typing (219) of quantum measurements enhances to mixed states, by first applying $\mathtt{collapse}_W$ (219) to states and co-states in parallel, and then $\bigcirc$–$\mathtt{pairing}$ (210) the result, whence we may and will denote this operation by the same symbol "$\mathtt{collapse}_W$":

mixed states **density matrices** **measure separately states and co-states** **decohere: discard off-diagonal entries** **probability distributions**

$$\begin{tikzcd}[column sep=large]
\begin{matrix}\mathrm{Q}W\\ \otimes\\ (\mathrm{Q}W)^*\end{matrix} \arrow[r, "\simeq"] \arrow[rrrr, bend right, "\mathtt{collapse}_W"'] & \begin{matrix}\oplus_W\mathbb{C}\\ \otimes\\ \oplus_W\mathbb{C}^*\end{matrix} \arrow[r, "\mathtt{collapse}_w \equiv \mathrm{join}^{\bigcirc_W}_{\mathbb{C}} \circ \mathrm{ret}^{\bigcirc_W}_{\oplus_W\mathbb{C}}", "\mathtt{collapse}_w \equiv \mathrm{join}^{\bigcirc_W}_{\mathbb{C}^*} \circ \mathrm{ret}^{\bigcirc_W}_{\oplus_W\mathbb{C}^*}"', "\otimes" description] & \begin{matrix}\bigcirc_W\mathbb{C}\\ \otimes\\ \bigcirc_W\mathbb{C}^*\end{matrix} \arrow[r, "\mathrm{pair}^{\bigcirc_W}_{\mathbb{C},\mathbb{C}^*}" description] & \bigcirc_W \begin{matrix}\mathbb{C}\\ \otimes\\ \mathbb{C}^*\end{matrix} \arrow[r, "\bigcirc_W\,\mathrm{ev}" description] & \bigcirc_W\mathbb{C}
\end{tikzcd}$$

tensor product of free $\bigcirc_W$-modales

separately handle pure $\bigcirc_W$- effects

monoidal monad structure on $\bigcirc_W$

$$|\psi\rangle\otimes\langle\psi| \;\mapsto\; \begin{matrix}\big(\sum_w |w\rangle\langle w|\psi\rangle\big)\\ \otimes\\ \big(\sum_{w'}\langle\psi|w'\rangle\langle w'|\big)\end{matrix} \;\mapsto\; \Big((w,w') \mapsto |w\rangle\underbrace{\langle w|\psi\rangle\langle\psi|w'\rangle}_{\text{coherences}}\langle w'|\Big) \;\mapsto\; \big(w \;\mapsto\; \big|\langle w|\psi\rangle\big|^2\big) \qquad (221)$$

a pure state among mixed

Born rule

Noteworthy are two remarkable aspects of this $\bigcirc_W$-effectful map:

- **(i)** in this form, the coherent quantum phases drop out, as expected for a realistic quantum measurement (the failure of which to happen for the analogous process on pure states was highlighted in [CPaq08, §1.6], where a different solution was discussed),
- **(ii)** in fact, (221) reproduces exactly the typing of the quantum measurement process in *Lüders' first form* (44), neatly embodying the Born rule (32).

In conclusion: Due to the symmetric monoidal monad structure on the indefiniteness-modality $\bigcirc$, the monadic typing of classically controlled quantum circuits with dynamically lifted quantum measurement gates has *syntactically* the same form whether applied to pure or to mixed states.

The difference with interpreting quantum circuits in the generality of mixed states is that here further stochastic quantum operations become available, the *quantum channels*. We discuss this in §2.5.

2.5 Mixed Quantum Types

We discuss a natural monadic formalization of mixed quantum states (35) and their quantum channels (39). The key observation is once again that the main structure happens to come for free as (co)monadic (co)effects that need not be postulated but are definable (admissible) in a suitably expressive linear type theory:

(i) quantum channel dynamics (39) on mixed quantum states (35) and their quantum observables (60) is all encoded by *transformations* (103) of the QuantumState (co)monads $\mathcal{H}$State

(ii) the collapsing measurement process on such mixed states is given by the monoidal monadic structure on the $\bigcirc$-modality (221).

What requires a little extra work to formalize is, finally:

(iii) the dagger-structure $(-)^\dagger$ (34) on quantum types. This has a rather beautiful (homotopy-)type-theoretic solution, which however is beyond the scope of this article and instead relegated to [SS23-QR].

For the present discussion, we assume the existence of operator adjoints as a black box; in fact, we exclusively need *dual* operator adjoints.

$$\mathcal{H}_1, \mathcal{H}_2 \;:\; \mathrm{QuType}^{\mathrm{fdm}}, \qquad f : \mathcal{H}_1 \to \mathcal{H}_2 \qquad \vdash \qquad f^{\dagger *} \;:\; \mathcal{H}_1^* \to \mathcal{H}_2^* \,. \tag{222}$$

We find that the structure of quantum probability theory (Lit. 1.12) — where quantum gates operating on pure quantum states are generalized to quantum channels operating on mixed quantum states (density matrixes) — is closely reflected in the monadic computational theory (Lit. 1.17) of the linear analog of the classical State/Store (co)monads, namely the QuantumState Frobenius monads $\mathcal{H}\mathrm{State} \equiv (\text{-}) \otimes \mathcal{H} \otimes \mathcal{H}^*$

Quantum Probability Theory	**QuantumState monadic computation**
Quantum channels	QuantumState transformations
Mixed quantum states	QuantumState effectful scalars
Quantum observables	QuantumState contextful scalars
Evolution of quantum observables	QuantumState transformation on modales

QuantumState modality. We consider the evident linear version of the classical state monad (83) and the classical store comonad (118), which over a *finite*-dimensional quantum state space fuse to a Frobenius monad (122) that, we will see, quite deserves to be called the *QuantumState modality*.

Definition 2.43 (QuantumState). For $\mathcal{H} \;:\; \mathrm{QuType}^{\mathrm{fdm}}$ a strongly dualizable linear type (133) (hence a finite-dimensional vector space in the model of Def. 2.1) with dual $\mathcal{H}^* \simeq \mathcal{H} \multimap \mathbb{1}$ (137), we say that the corresponding *QuantumState* (co)monads are the Frobenius monads (122) induced (74) by the corresponding ambidextrous adjunction of tensoring functors (138):

QuantumState
Frobenius
monad

$$\begin{tikzcd}
\mathrm{QuType} \arrow[loop above, "\mathcal{H}\mathrm{State}"] \arrow[loop below, "\mathcal{H}^*\mathrm{Store}"] \arrow[r, bend left, "(\text{-})\otimes\mathcal{H}"] \arrow[r, bend right, "(\text{-})\otimes\mathcal{H}"'] & \mathrm{QuType} \arrow[l, "(\text{-})\otimes\mathcal{H}^*" description] \arrow[loop above, "\mathcal{H}\mathrm{Store}"] \arrow[loop below, "\mathcal{H}^*\mathrm{State}"]
\end{tikzcd} \tag{223}$$

$\perp \quad \perp \quad \perp \quad \perp$

Since the ambidexterity means that $\mathcal{H}$State and $\mathcal{H}^*$Store fuse to a single Frobenius monad (122), we will often refer to both or either as *QuantumState modalities* and speak of the *QuantumStore modality* when referring specifically only to the comonad structure.

For the record, in bra-ket notation (28) the return/obtain-operations of QuantumState are as follows (where $W, QW \simeq \mathcal{H}$ denotes any orthonormal basis for $\mathcal{H}$ with respect to any chose Hermitian inner product $\langle \cdot | \cdot \rangle$):

$$\begin{array}{|c|c|}
\hline
\begin{array}{ccc}
|\kappa\rangle\,|\psi\rangle & \longmapsto & |\kappa\rangle\,|\psi\rangle \\
\mathcal{K} \otimes \mathcal{H} & \xrightarrow{\qquad} & \mathcal{K} \otimes \mathcal{H} \\
\hline
\mathcal{K} & \xrightarrow{\mathrm{ret}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}}} & \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \\
|\kappa\rangle & \longmapsto & \sum\limits_{w} |\kappa\rangle \otimes |w\rangle\langle w|
\end{array}
&
\begin{array}{ccc}
|\kappa\rangle\,\langle\phi| & \longmapsto & |\kappa\rangle\,\langle\phi| \\
\mathcal{K} \otimes \mathcal{H}^* & \xrightarrow{\mathrm{id}} & \mathcal{K} \otimes \mathcal{H}^* \\
\hline
\mathcal{K} \otimes \mathcal{H}^* \otimes \mathcal{H} & \xrightarrow{\mathrm{obt}^{\mathcal{H}\mathrm{Store}}_{\mathcal{K}}} & \mathcal{K} \\
|\kappa\rangle\,\langle\phi| \otimes |\psi\rangle & \longmapsto & |\kappa\rangle\langle\phi|\psi\rangle
\end{array}
\\
\hline
\begin{array}{ccc}
|\kappa\rangle\,|\psi\rangle & \longmapsto & |\kappa\rangle\,|\psi\rangle \\
\mathcal{K} \otimes \mathcal{H} & \xrightarrow{\mathrm{id}} & \mathcal{K} \otimes \mathcal{H} \\
\hline
\mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* & \xrightarrow{\mathrm{obt}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}}} & \mathcal{K} \\
|\kappa\rangle\,|\psi\rangle\langle\phi| & \longmapsto & |\kappa\rangle\langle\phi|\psi\rangle
\end{array}
&
\begin{array}{ccc}
|\kappa\rangle\,\langle\phi| & \longmapsto & |\kappa\rangle\,\langle\phi| \\
\mathcal{K} \otimes \mathcal{H}^* & \xrightarrow{\mathrm{id}} & \mathcal{K} \otimes \mathcal{H}^* \\
\hline
\mathcal{K} & \xrightarrow{\mathrm{ret}^{\mathcal{H}^*\mathrm{State}}_{\mathcal{K}}} & \mathcal{K} \otimes \mathcal{H}^* \otimes \mathcal{H} \\
|\kappa\rangle & \longmapsto & \sum\limits_{w} |\kappa\rangle\langle w| \otimes |w\rangle
\end{array}
\\
\hline
\end{array} \tag{224}$$

so that the join/duplicate-operations are as follows:

$$
\begin{array}{|c|c|}
\hline
\begin{array}{ccc}
\mathcal{K}\otimes\mathcal{H}^*\otimes\mathcal{H} & \xrightarrow[\mathrm{ret}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}\otimes\mathcal{H}^*}\otimes\mathcal{H}]{\mathrm{dupl}^{\mathcal{H}\mathrm{Store}}_{\mathcal{K}}} & \mathcal{K}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H} \\
|\kappa\rangle\,\langle\phi|\otimes|\psi\rangle & \mapsto & \sum\limits_{w}|\kappa\rangle\,\langle\phi|\otimes|w\rangle\langle w|\otimes|\psi\rangle
\end{array}
&
\begin{array}{ccc}
\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow[\mathrm{obt}^{\mathcal{H}\mathrm{Store}}_{\mathcal{K}\otimes\mathcal{H}}\otimes\mathcal{H}^*]{\mathrm{join}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}}} & \mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^* \\
|\kappa\rangle\otimes|-\rangle\langle\phi|\otimes|\psi\rangle\langle-| & \mapsto & |\kappa\rangle\langle\phi|\psi\rangle\otimes|-\rangle\langle-|
\end{array}
\\
\hline
\begin{array}{ccc}
\mathcal{K}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H} & \xrightarrow[\mathrm{obt}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}\otimes\mathcal{H}^*}]{\mathrm{join}^{\mathcal{H}^*\mathrm{State}}_{\mathcal{K}}} & \mathcal{K}\otimes\mathcal{H}^*\otimes\mathcal{H} \\
|\kappa\rangle\,\langle-|\otimes|\psi\rangle\langle\phi|\otimes|-\rangle & \mapsto & \langle\phi|\psi\rangle\langle-|\otimes|-\rangle
\end{array}
&
\begin{array}{ccc}
\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow[\mathrm{ret}^{\mathcal{H}^*\mathrm{State}}_{\mathcal{K}\otimes\mathcal{H}}\otimes\mathcal{H}^*]{\mathrm{dupl}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}}} & \mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^* \\
|\kappa\rangle\otimes|\psi\rangle\langle\phi| & \mapsto & \sum\limits_{w}|\kappa\rangle|\psi\rangle\langle w|\otimes|w\rangle\langle\phi|
\end{array}
\\
\hline
\end{array}
\tag{225}
$$

Notice that these operations all express in one way or another the basic bra-ket manipulations known from quantum mechanics textbooks (evaluation and "insertion of an identity"). In particular, the zig-zag identities which witness the adjunctions in (223) are nothing but the following familiar basic identities:

$$
\begin{array}{rcccc}
\mathtt{obtain}^{\mathcal{H}\mathrm{Store}}_{\mathcal{K}\otimes\mathcal{H}}\circ(\mathtt{return}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}}\otimes\mathcal{H})(|\kappa\rangle\otimes|\psi\rangle) & \equiv & |\kappa\rangle\otimes\sum\limits_{w}|w\rangle\langle w|\psi\rangle & = & |\kappa\rangle\otimes|\psi\rangle \\
(\mathtt{obtain}^{\mathcal{H}\mathrm{Store}}_{\mathcal{K}}\otimes\mathcal{H})\circ\mathtt{return}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}\otimes\mathcal{H}^*}(|\kappa\rangle\otimes\langle\phi|) & \equiv & |\kappa\rangle\otimes\sum\limits_{w}\langle\phi|w\rangle\langle w| & = & |\kappa\rangle\otimes\langle\phi|\,.
\end{array}
$$

Remark 2.44 (QuantumState as QuantumWriter). The QuantumState Frobenius monad of Def. 2.43 is equivalently the linear (co)Writer monad (81) over $\mathcal{H}\otimes\mathcal{H}^*$, the latter understood with its canonical Frobenius monoid structure of endomorphism objects in compact closed categories (see e.g. [Vic11, Lem. 3.17]):

$$
\begin{array}{ccc}
\textbf{quantum state} & \textbf{quantum (co)writer} & \textbf{quantum store} \\
\mathcal{H}\mathrm{State} & (\text{-})\otimes(\mathcal{H}\otimes\mathcal{H}^*) & \mathcal{H}^*\mathrm{Store}
\end{array}
$$

$$
\begin{tikzcd}
\mathrm{Monads} & \mathrm{FrobMonads} \arrow[l] \arrow[r] & \mathrm{CoMonads}
\end{tikzcd}
$$

In particular, if $\mathcal{H}\simeq QW$ then QuantumState is the (co)Writer monad for $QW\otimes QW^*$, in which form it is interesting to compare to the quantum indefiniteness/randomness modality, which is the (co)writer for a single copy QW, according to Prop. 2.35.

Frobenius algebra	**Quantum modalities**	**Quantum effects**
QW	indefiniteness/randomness	collapsing quantum measurement
$QW\otimes QW^*$	quantum state/store	quantum probability

Proposition 2.45 (QuantumState effect-/contextful maps are Linear operators).
(i) *The* $\mathcal{H}^*$State *modality of Def. 2.43 has (co)Kleisli morphisms of the form*

$$
\begin{array}{ccccc}
\begin{array}{ccc}
\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow{O_A} & \mathcal{L} \\
|\kappa\rangle\otimes|\psi\rangle\langle\phi| & \mapsto & \langle\phi,-|\,A\,|\kappa,\psi\rangle
\end{array}
& \leftrightarrow & A\;:\mathcal{K}\otimes\mathcal{H}\to\mathcal{L}\otimes\mathcal{H} & \leftrightarrow &
\begin{array}{ccc}
\mathcal{K} & \xrightarrow{\mathcal{S}_A} & \mathcal{L}\otimes\mathcal{H}\otimes\mathcal{H}^* \\
|\kappa\rangle & \mapsto & \langle-,-|\,A\,|\kappa,-\rangle
\end{array}
\end{array}
\tag{226}
$$

(ii) *on which the bind/extend- operations are given by*

$$
\begin{array}{lcl}
\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K},\mathcal{L}} & : & (\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\multimap\mathcal{L})\multimap(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\multimap\mathcal{L}\otimes\mathcal{H}\otimes\mathcal{H}^*) \\
\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K},\mathcal{L}} & \equiv & (|\kappa\rangle\,|\psi\rangle\langle\phi|\;\mapsto\;\langle\phi,-|A|\kappa,\psi\rangle)\;\mapsto\;(|\kappa\rangle\,|\psi\rangle\langle\phi|\;\mapsto\;A|\kappa,\psi\rangle\langle\phi|)
\end{array}
\tag{227}
$$

$$
\begin{array}{lcl}
\mathtt{bind}^{\mathcal{H}\mathrm{State}}_{\mathcal{K},\mathcal{L}} & : & (\mathcal{K}\multimap\mathcal{L}\otimes\mathcal{H}\otimes\mathcal{H}^*)\multimap(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\multimap\mathcal{L}\otimes\mathcal{H}\otimes\mathcal{H}^*) \\
\mathtt{bind}^{\mathcal{H}\mathrm{State}}_{\mathcal{K},\mathcal{L}} & \equiv & (|\kappa\rangle\;\mapsto\;\langle-,-|\,A\,|\kappa,-\rangle)\;\mapsto\;(|\kappa\rangle|\psi\rangle\langle\phi|\;\mapsto\;A\,|\kappa,\psi\rangle\langle\phi|)
\end{array}
\tag{228}
$$

(iii) *Hence we have bijections*

$$
\begin{array}{ccccc}
\mathcal{H}\mathrm{State}\textbf{-contextful maps} & & \textbf{linear operators} & & \mathcal{H}\mathrm{State}\textbf{-effectful maps} \\
O_A\;:\;\mathcal{H}\mathrm{State}(\mathcal{K})\to\mathcal{L} & \leftrightarrow & A\;:\;\mathcal{K}\otimes\mathcal{H}\to\mathcal{L}\otimes\mathcal{H} & \leftrightarrow & \mathcal{S}_A\;:\;\mathcal{K}\to\mathcal{H}\mathrm{State}(\mathcal{L})
\end{array}
$$

under which Kleisli composition corresponds to ordinary composition of linear operators:

$$
\begin{array}{ccccc}
\textbf{composition of } \mathcal{H}\mathrm{State}\textbf{-contextful maps} & & & & \textbf{composition of } \mathcal{H}\mathrm{State}\textbf{-effectful maps} \\
O_A\circ(\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}(O_B)) & = & O_{A\cdot B}\;\;\leftrightarrow\;\;\mathcal{S}_{A\cdot B} & = & \mathcal{S}_A\circ(\mathtt{bind}^{\mathcal{H}\mathrm{State}}(\mathcal{S}_B))\,. \\
 & & \textbf{composition of linear operators} & &
\end{array}
$$

Proof. By direct unwinding of the formulas (73) and (225):

$$
\begin{array}{rcccccc}
 & \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* & \xrightarrow{\mathrm{dupl}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}^*}} & \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^* & \xrightarrow{O_A \otimes \mathcal{H} \otimes \mathcal{H}^*} & \mathcal{L} \otimes \mathcal{H} \otimes \mathcal{H}^* \\
\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K},\mathcal{L}} O_A \;: & |\kappa\rangle \otimes |\psi\rangle\langle\phi| & \mapsto & \sum_w |\kappa\rangle\, |\psi\rangle\langle w| \otimes |w\rangle\langle\phi| & \mapsto & \sum_w |w\rangle\langle w, -|\, A\, |\kappa, \psi\rangle\langle\phi| \\
 & & & & = & A\, |\kappa, \psi\rangle\langle\phi|
\end{array}
$$

$$
\begin{array}{rcccccc}
 & \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* & \xrightarrow{\mathcal{S}_A \otimes \mathcal{H} \otimes \mathcal{H}^*} & \mathcal{L} \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^* & \xrightarrow{\mathrm{join}^{\mathcal{H}\mathrm{State}}_{\mathcal{L}}} & \mathcal{L} \otimes \mathcal{H} \otimes \mathcal{H}^* \\
\mathtt{bind}^{\mathcal{H}\mathrm{State}}_{\mathcal{K},\mathcal{L}} \mathcal{S}_A \;: & |\kappa\rangle \otimes |\psi\rangle\langle\phi| & \mapsto & \langle -, -|\, A\, |\kappa, -\rangle \otimes |\psi\rangle\langle\phi| & \mapsto & A\, |\kappa, \psi\rangle\langle\phi|
\end{array}
\qquad \square
$$

Quantum observables. We show that the core structure of *quantum observables* is reflected in the QuantumState-contextful scalars (Ex. 2.46) including:

- their expectation values (229),
- their algebra structure (230),
- their Heisenberg-evolution (Prop. 2.50).

Example 2.46 (Quantum observables are the QuantumState contextful scalars). Notice that in any monoidal category like $(\mathrm{QuType}, \otimes, \mathbb{1})$ it makes sense to refer to the endomorphisms $c : \mathbb{1} \to \mathbb{1}$ of the tensor unit as the *scalars* of the theory ([AC04, §6][HV12, 2.1]). Therefore, with the understanding of comonadic computational contexts (Lit. 1.17) and given a comonad C on QuType, the Kleisli-endomorphisms of the tensor unit $C(\mathbb{1}) \to \mathbb{1}$ may be thought of (116) as the *C-contextful scalars*. Now Prop. 2.45 says that the $\mathcal{H}$State-*contextful scalars* are equivalently the linear operators on $\mathcal{H}$, here seen to be representing quantum observables (60) incarnated via their system of expectation values (61):

$$
\begin{array}{ccc}
O_A \;: \mathcal{H} \otimes \mathcal{H}^* & \longrightarrow & \mathbb{1} \\
|\psi\rangle\langle\phi| & \mapsto & \langle\phi|\, A\, |\psi\rangle \\
\rho & \mapsto & \mathrm{Tr}(\rho \cdot A)
\end{array}
\quad \leftrightarrow \quad A \;:\; \mathcal{H} \to \mathcal{H}\,. \tag{229}
$$

Moreover, the (Kleisli-)composition of such QuantumState-contextful scalars reproduces the ordinary operator product of the corresponding linear operators:

$$
O_A \circ \mathtt{extend}^{\mathcal{H}\mathrm{Store}}_{\mathbb{1}} O_B \;=\; O_{A \cdot B}\,, \qquad \text{so that} \qquad \overset{\substack{\textbf{QuantumState}\\ \textbf{Kleisli-endomorphism}\\ \textbf{algebra of tensor unit}}}{\mathrm{QuType}_{\mathcal{H}\mathrm{Store}}(\mathbb{1}, \mathbb{1})} \;\simeq\; \overset{\substack{\textbf{algebra of}\\ \textbf{linear operators}}}{\mathrm{End}(\mathcal{H})} \quad (\text{as algebras})\,. \tag{230}
$$

Remark 2.47 (The operational/logical meaning of operator products of quantum observables).
(i) It is commonplace in modern quantum physics that the *algebra of quantum observables* is indeed that: an associative algebra under operator products of the corresponding linear operators. However, while mathematically suggestive, it is subtle to decide which aspect of quantum reality is really modeled by forming the plain operator product of a pair of *non-commuting* observables $O_A, O_{A'}$; because in this case a prescription for measuring them separately (namely via their respective eigenbases W, W') does not readily yield a prescription for measuring their operator product O_{AB}.
(ii) This issue was felt to be severe enough of a conceptual problem by the founding fathers of quantum physics that another non-associative notion of algebras of quantum observables was proposed [Jor32][JvNW34], now known as *Jordan algebras* (see [Ba20] for more on the quantum foundational motivation of Jordan algebras). However, while the concept of Jordan algebras turned out to be useful in various areas of mathematics, its relevance for conceptualizing quantum observables has remained inconclusive.
(iii) Indeed, the highly successful modern algebraic formulation of quantum physics (for a good exposition see [Gl09][Gl11]) is entirely based on the associative algebra structure on observables (further promoted to a C^*-algebra structure for infinite-dimensional algebras) and has no use of Jordan algebras.
(iv) This begs the question that may originally have motivated Jordan et al.: To give a *logical* justification from first principles for considering quantum observables as an associative algebra under operator products. But if we grant (with Lit. 1.4, 1.13 and 1.17) a foundational logical content to natural (co)monadic structures on linear types, then Ex. 2.46 provides a satisfactory answer.

For the following Proposition 2.48, recall (Lit. 1.12) that for a pair of quantum systems (represented by) $\mathcal{H}_1, \mathcal{H}_2$: QuType, a *quantum channel* (39) between them is a (linear) map of the form

$$
\mathcal{H}_1 \otimes \mathcal{H}_1^* \xrightarrow{\;\mathrm{chan}\;} \mathcal{H}_2 \otimes \mathcal{H}_2^*
$$

satisfying some properties; and that in general such a channel may act among further "ancillary" systems $\mathcal{K}$ (such as $\mathcal{K} = \mathcal{B} \otimes \mathcal{B}^*$, for $\mathcal{B}$ a "bath" environment), being more generally a tensor map of the form

$$
\mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \xrightarrow{\;\mathrm{id}_{\mathcal{K}} \otimes \mathrm{chan}\;} \mathcal{K} \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^*\,.
$$

Proposition 2.48 (Unitary quantum channels are quantum state transformations). *The unitary quantum channel* $U \otimes U^{\dagger^*}$ (40) *corresponding to a unitary operator*[33] $U : \mathcal{H}_1 \to \mathcal{H}_2$ *induces a (co)monad transformation* (107) *between the*

---

[33] For the statement of the proposition at this point it just matters that U is an invertible linear map with inverse denoted $U^\dagger$.

corresponding Quantum State (co)monads, in that

QuantumState transformation

$$\begin{tikzcd}
\mathcal{H}_1\mathrm{State} \arrow[rr, "\mathrm{chan}^U"] && \mathcal{H}_2\mathrm{State} \\
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rr, "(-)\otimes U\otimes U^{\dagger *}"] && (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \\
|\kappa\rangle\otimes\rho \arrow[rr, mapsto] && |\kappa\rangle\otimes(U\cdot\rho\cdot U^\dagger)
\end{tikzcd}$$

unitary quantum channel

Proof. We need to check the compatibility conditions (108). But since the (co)unit of $\mathcal{H}$State is given (224) by inserting an identity and by inner product, respectively, their preservation is essentially the definition of two-sided inverse operators. As a warmup for the following computations, we spell this out.

$$\begin{tikzcd}
(\text{-}) \arrow[rrr, equal] \arrow[dddd, "\mathrm{ret}^{\mathcal{H}_1\mathrm{State}}_{(\text{-})}"'] &&& (\text{-}) \arrow[dddd, "\mathrm{ret}^{\mathcal{H}_2\mathrm{State}}_{(\text{-})}"] \\
& |\text{-}\rangle \arrow[r, mapsto] \arrow[dd, mapsto] & |\text{-}\rangle \arrow[d, mapsto] & \\
& & |\text{-}\rangle\otimes\sum\limits_{w_2}|w_2\rangle\langle w_2| \arrow[d, equal, "(231)"'] & \\
& |\text{-}\rangle\otimes\sum\limits_{w_1}|w_1\rangle\langle w_1| \arrow[r, mapsto] & |\text{-}\rangle\otimes\sum\limits_{w_1}U|w_1\rangle\langle w_1|U^\dagger & \\
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrr, "(-)\otimes U\otimes U^{\dagger *}"'] &&& (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*
\end{tikzcd}$$

$$\begin{tikzcd}
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrr, "(-)\otimes U\otimes U^{\dagger *}\otimes U\otimes U^{\dagger *}"] \arrow[dddd, "\mathrm{join}^{\mathcal{H}_1\mathrm{State}}_{(\text{-})}"'] &&& (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[dddd, "\mathrm{join}^{\mathcal{H}_2\mathrm{State}}_{(\text{-})}"] \\
& |\text{-}\rangle\otimes|\text{-}\rangle\langle\phi|\otimes|\psi\rangle\langle\text{-}| \arrow[r, mapsto] \arrow[dd, mapsto] & |\text{-}\rangle\otimes U|\text{-}\rangle\langle\psi|U^\dagger\otimes U|\psi\rangle\langle\text{-}|U \arrow[d, mapsto] & \\
& & |\text{-}\rangle\otimes U|\text{-}\rangle\langle\phi|U^\dagger U|\psi\rangle\langle\text{-}|U^\dagger \arrow[d, equal] & \\
& |\text{-}\rangle\otimes|\text{-}\rangle\langle\phi|\psi\rangle\langle\text{-}| \arrow[r, mapsto] & |\text{-}\rangle\otimes U|\text{-}\rangle\langle\phi|\psi\rangle\langle\text{-}|U^\dagger & \\
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrr, "(-)\otimes U\otimes U^{\dagger *}"'] &&& (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*
\end{tikzcd}$$

$$\begin{tikzcd}
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrr, "(-)\otimes U\otimes U^{\dagger *}"] \arrow[dddd, "\mathrm{obt}^{\mathcal{H}_1^*\mathrm{Store}}_{(\text{-})}"'] &&& (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[dddd, "\mathrm{obt}^{\mathcal{H}_2^*\mathrm{Store}}_{(\text{-})}"] \\
& |\text{-}\rangle\otimes|\psi\rangle\langle\phi| \arrow[r, mapsto] \arrow[dd, mapsto] & |\text{-}\rangle\otimes U|\psi\rangle\langle\phi|U^\dagger \arrow[d, mapsto] & \\
& & |\text{-}\rangle\otimes\langle\phi|U^\dagger U|\psi\rangle \arrow[d, equal] & \\
& |\text{-}\rangle\,\langle\phi|\psi\rangle \arrow[r, mapsto] & |\text{-}\rangle\,\langle\phi|\psi\rangle & \\
(\text{-}) \arrow[rrr, equal] &&& (\text{-})
\end{tikzcd}$$

$$\begin{tikzcd}
(\text{-})\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrr, "(-)\otimes U\otimes U^{\dagger *}"] \arrow[dddd, "\mathrm{dupl}^{\mathcal{H}_1^*\mathrm{Store}}_{(\text{-})}"'] &&& (\text{-})\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[dddd, "\mathrm{dupl}^{\mathcal{H}_2^*\mathrm{Store}}_{(\text{-})}"] \\
& |\text{-}\rangle\otimes|\psi\rangle\langle\phi| \arrow[r, mapsto] \arrow[dd, mapsto] & |\text{-}\rangle\otimes U|\psi\rangle\langle\phi|U^\dagger \arrow[d, mapsto] & \\
& & |\text{-}\rangle\otimes\sum\limits_{w}U|\psi\rangle\langle w|\otimes|w\rangle\langle\phi|U^\dagger \arrow[d, equal, "(232)"'] & \\
& |\text{-}\rangle\otimes\sum\limits_{w}|\psi\rangle\langle w|\otimes|w\rangle\langle\phi| \arrow[r, mapsto] & |\text{-}\rangle\otimes\sum\limits_{w}U|\psi\rangle\langle w|U^\dagger\otimes U|w\rangle\langle\phi|U^\dagger & \\
(\text{-}) \arrow[rrr, equal] &&& (\text{-})
\end{tikzcd}$$

□

Here and in the following we make repeated use of the following elementary but important relations for linear maps $E : \mathcal{H}_1 \to \mathcal{H}_2$:

$$\begin{tikzcd}
\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[rr, "\sim"] && (\mathcal{H}_2 \multimap \mathcal{H}_2) \\
\sum\limits_{w} E|w\rangle\langle w|E^\dagger \;=\; E\Big(\sum\limits_{w}|w\rangle\langle w|\Big)E^\dagger \arrow[rr, mapsto] && E\cdot\mathrm{id}_{\mathcal{H}_1}\cdot E^\dagger \;=\; E\cdot E^\dagger
\end{tikzcd} \tag{231}$$

$$\begin{tikzcd}
\mathcal{H}_2^* \otimes \mathcal{H}_2 \arrow[rr, "\sim"] & & (\mathcal{H}_2^* \multimap \mathcal{H}_2^*) \\
\sum\limits_w \langle w | E^\dagger \otimes E | w \rangle \;=\; \sum\limits_w \langle w | E^\dagger \otimes E | w \rangle \arrow[rr, mapsto] & & (E \cdot E^\dagger)^*
\end{tikzcd} \tag{232}$$

The following Prop. 2.50 invokes the covariant action (106) of monad transformations (107) on free modales, but restricted to the special case where the monad transformation is an isomorphism. In order to amplify the canonicity of this construction, the following Lemma 2.49 highlights that in this case the transformation is equal to the inverse of the *contra*variant action (111) of monad morphisms of general modales (which is more commonly discussed in the monad-literature), restricted to free modales.

Lemma 2.49 (Evolution of free modales along isomorphic transformations of monads).
(i) *On* isomorphic *monad transformation,* trans : $\mathcal{E} \xrightarrow{\sim} \mathcal{E}'$ (107)*, the induced contravariant functor* trans* (111) *on general modales is naturally isomorphic to the inverse* $\circ$trans^{-1} *of the induced covariant functor* (106) *on free modales* (93)*, via the natural isomorphism whose components are just the components* trans$_{(\text{-})}$ *of the natural transformation* trans*:*

$$\begin{tikzcd}
\mathcal{E}' & \mathcal{E} \arrow[l, "\text{trans}"', "\sim"] \\
\mathrm{Type}_{\mathcal{E}'} \arrow[r, "\text{frtrans}^*", "\sim"'] \arrow[d, hook] & \mathrm{Type}_{\mathcal{E}} \arrow[d, hook] \arrow[dl, Rightarrow, "\text{trans}_{(\text{-})}", "\sim"'] \\
\mathrm{Type}^{\mathcal{E}'} \arrow[r, "\sim", "\text{trans}^*"'] & \mathrm{Type}^{\mathcal{E}}
\end{tikzcd} \tag{233}$$

(ii) *In that on Kleisli morphisms* (95) *this is given by postcomposition with the inverse transformation* trans$^{-1}_{(\text{-})}$ *and as such*

$$\text{frtrans}^*\mathtt{bind}^{\mathcal{E}'}\Big(D_1 \xrightarrow{f'} \mathcal{E}'(D_2)\Big) \;\;=\;\; \mathtt{bind}^{\mathcal{E}}\Big(D_1 \xrightarrow{f'} \mathcal{E}'(D_2) \xrightarrow{\text{trans}^{-1}_{D_2}} \mathcal{E}(D_2)\Big). \tag{234}$$

Proof. First, notice the following diagram, which commutes by the defining properties of trans (108) and the very definition of trans* (111).

$$D : \mathrm{Type} \quad \vdash \quad \begin{tikzcd}
\mathcal{E}\mathcal{E}(D) \arrow[r, "\mathcal{E}(\text{trans}_D)", "\sim"'] \arrow[d, "\text{join}^{\mathcal{E}}_D"] & \mathcal{E}\mathcal{E}'(D) \arrow[r, "\text{trans}_{\mathcal{E}'(D)}"] \arrow[d, "\rho \equiv \text{trans}^*\rho'"] & \mathcal{E}'\mathcal{E}'(D) \arrow[d, "\rho' \equiv \text{join}^{\mathcal{E}'}_D"] \\
\mathcal{E}(D) \arrow[r, "\sim", "\text{trans}_D"'] & \mathcal{E}'(D) \arrow[r, equal] & \mathcal{E}'(D) \\[-20pt]
\textbf{free } \mathcal{E}\textbf{-modale} & \textbf{isomorphic to} \quad \textbf{transformation of} & \textbf{free } \mathcal{E}'\textbf{-modale}
\end{tikzcd} \tag{235}$$

But the left square now exhibits trans$_D$: $\mathcal{E}(D) \xrightarrow{\sim}$ trans$^*\mathcal{E}'(D)$ as a homomorphism of modales (91) from the free $\mathcal{E}$-modale on D to the transformation of the free $\mathcal{E}'$-modale on D; and this homomorphism is an isomorphism by the assumption that trans is an isomorphism, as shown. Therefore the claimed natural transformation in (233) is given in components as follows:

$$\begin{tikzcd}
\mathrm{Type}_{\mathcal{E}'} \arrow[rr, bend left, "\text{frtrans}^*"{name=U}] \arrow[rr, bend right, "\text{trans}^*"'{name=L}] \arrow[from=U, to=L, Rightarrow, "\text{trans}_{(\text{-})}"] & & \mathrm{Type}^{\mathcal{E}} \\
\mathcal{E}'(D_1) \arrow[ddd, "\phi"] & \mapsto & \mathcal{E}(D_1) \arrow[r, "\text{trans}_{D_1}", "\sim"'] \arrow[d, "\text{trans}_{D_1}"] & \mathcal{E}'(D_1) \arrow[ddd, "\phi"] \\
& & \mathcal{E}'(D_1) \arrow[d, "\phi"] & \\
& & \mathcal{E}'(D_2) \arrow[d, "\text{trans}^{-1}_{D_2}"] & \\
\mathcal{E}'(D_2) & \mapsto & \mathcal{E}(D_2) \arrow[r, "\sim", "\text{trans}_{D_2}"'] & \mathcal{E}'(D_2)
\end{tikzcd} \tag{236}$$

From this, we get the following commuting diagram, where the left square commutes by the transformation property (104) while the right square commutes by (236):

$$\begin{tikzcd}
D_1 \arrow[r, "\text{ret}^{\mathcal{E}}_{D_1}"] \arrow[d, equal] & \mathcal{E}(D_1) \arrow[r, "\text{frtrans}^*\mathtt{bind}^{\mathcal{E}'} f'"] \arrow[d, "\text{trans}_{D_1}"] & \mathcal{E}(D_2) \\
D_1 \arrow[r, "\text{ret}^{\mathcal{E}'}_{D_1}"'] \arrow[rr, bend right, "f'"'] & \mathcal{E}'(D_1) \arrow[r, "\mathtt{bind}^{\mathcal{E}'} f'"'] & \mathcal{E}'(D_2) \arrow[u, "\text{trans}^{-1}_{D_2}"']
\end{tikzcd}$$

and the claim (234) is the image under $\mathtt{bind}^{\mathcal{E}}$ of this equality. □

As we apply (in Prop. 2.50) Lem. 2.49 to QuantumStore-contextful maps, hence to Kleisli maps for a *co*monad, beware that the role of covariant and contravariant functors gets interchanged.

Proposition 2.50 (QuantumState evolution is Heisenberg evolution). *For $U : \mathcal{H}_1 \to \mathcal{H}_2$ a unitary linear map, the canonical evolution according to Lem. 2.49*

- *of quantum observables regarded a QuantumState-contextful scalars O_A (via Ex. 2.46)*
- *along the unitary quantum channel* chan^U *regarded as a QuantumState transformation (via Prop. 2.48)*

is Heisenberg evolution (62)

$$\begin{tikzcd}
\mathcal{H}_2 \otimes \mathcal{H}_2^* \arrow[r, "\mathrm{chan}^{U^{-1}}"] \arrow[rr, bend left, "O_{U\cdot A\cdot U^\dagger}"] & \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "O_A"] & \mathbb{1} \\
\rho \arrow[r, mapsto] & U^\dagger \cdot \rho \cdot U \arrow[r, mapsto] & \mathrm{tr}((U^\dagger \cdot \rho \cdot U) \cdot A) \\
 & {} \arrow[r, phantom, "="] & \mathrm{tr}(\rho \cdot (U \cdot A \cdot U^\dagger)) \\
 & {} \arrow[r, phantom, "="] & O_{U\cdot A\cdot U^\dagger}(\rho)\,.
\end{tikzcd} \tag{237}$$

Quantum channels as QuantumState transformations.

Proposition 2.51 (Uniform coupling channels are QuantumState transformations). *The quantum coupling channels to a uniform bath state* (56) *of some system $\mathcal{B}$*

$$\begin{tikzcd}
(\text{-}) \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\mathrm{id}\otimes\mathrm{ret}^{\mathcal{B}\mathrm{State}}_{\mathbb{1}}"] & (\text{-}) \otimes \mathcal{H} \otimes \mathcal{B} \otimes \mathcal{B}^* \otimes \mathcal{H} \arrow[r, "\sim"] & (\text{-}) \otimes (\mathcal{H} \otimes \mathcal{B}) \otimes (\mathcal{H} \otimes \mathcal{B})^* \\
|\text{-}\rangle \otimes |\psi\rangle\langle\psi'| \arrow[r, mapsto] & |\text{-}\rangle \otimes |\psi\rangle\Big(\sum_b |b\rangle\langle b|\Big)\langle\psi'| \arrow[r, phantom, "="] & |\text{-}\rangle \otimes \sum_b |\psi, b\rangle\langle b, \psi'|
\end{tikzcd}$$

are monadic QuantumState transformations

$$\mathrm{couple}^{\mathcal{B}} \;:\; \mathcal{H}\mathrm{State} \xrightarrow{\;\mathrm{mon}\;} (\mathcal{H} \otimes \mathcal{B})\mathrm{State}$$

and as such the components of a pointed endofunctor (110) *on* Mnd(QuType).

Proof. Since the structure maps of the $(\mathcal{H} \otimes \mathcal{B})$State-comonad are tensor products of structure maps of $\mathcal{H}$State and $\mathcal{B}$State, it is sufficient to show this for $\mathcal{H} = \mathbb{1}$, hence for the case that $\mathcal{H}\mathrm{State} = \mathrm{Id}$. But in this case $\mathrm{couple}^{\mathcal{B}} = \mathrm{ret}^{\mathcal{B}\mathrm{State}}_{(\text{-})}$, which we know to be a monadic transformation (in fact the initial one) according to (109).

Alternatively, it is immediate to explicitly check the required conditions. We have:

$$\begin{tikzcd}
(\text{-}) \arrow[rrr, equal] \arrow[dddd, "\mathrm{ret}^{\mathcal{H}\mathrm{State}}_{(\text{-})}"'] & & & (\text{-}) \arrow[dddd, "\mathrm{ret}^{(\mathcal{H}\otimes\mathcal{B})\mathrm{State}}_{(\text{-})}"] \\
 & |\text{-}\rangle \arrow[r, mapsto] \arrow[dd, mapsto] & |\text{-}\rangle \arrow[d, mapsto] & \\
 & & |\text{-}\rangle \otimes \sum_{w,b} |w, b\rangle\langle b, w| \arrow[d, equal] & \\
 & |\text{-}\rangle \otimes \sum_w |w\rangle\langle w| \arrow[r, mapsto] & |\text{-}\rangle \otimes \sum_{w,b} |w\rangle \otimes |b\rangle\langle b| \otimes \langle w| & \\
(\text{-}) \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[rrr, "\mathrm{couple}^{\mathcal{B}}"'] & & & (\text{-}) \otimes \mathcal{H} \otimes \mathcal{B} \otimes \mathcal{B}^* \otimes \mathcal{H}^*
\end{tikzcd}$$

$$\begin{tikzcd}
(\text{-}) \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[rrr, "\mathrm{couple}^{\mathcal{B}}_{(\cdots)} \circ (\mathrm{couple}^{\mathcal{B}} \otimes \mathrm{id})"] \arrow[dddd, "\mathrm{join}^{\mathcal{H}\mathrm{State}}_{(\text{-})}"'] & & & (\text{-}) \otimes \mathcal{H} \otimes \mathcal{B} \otimes \mathcal{B}^* \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{B} \otimes \mathcal{B}^* \otimes \mathcal{H}^* \arrow[dddd, "\mathrm{join}^{(\mathcal{H}\otimes\mathcal{B})\mathrm{State}}_{(\text{-})}"] \\
 & |\psi\rangle\langle\psi'| \otimes |\phi\rangle\langle\phi'| \arrow[r, mapsto] \arrow[dd, mapsto] & \sum_{b,b'} |\psi, b\rangle\langle b, \psi'| \otimes |\phi, b'\rangle\langle b', \phi'| \arrow[d, mapsto] & \\
 & & \langle\psi'|\phi\rangle \sum_{b,b'} |\psi, b\rangle\langle b|b'\rangle\langle b', \phi'| \arrow[d, equal] & \\
 & |\psi\langle\psi'|\phi\rangle\langle\phi'| \arrow[r, mapsto] & \langle\psi'|\phi\rangle \sum_b |\psi, b\rangle\langle b, \phi'| & \\
(\text{-}) \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[rrr, "\mathrm{couple}^{\mathcal{B}}"'] & & & (\text{-}) \otimes \mathcal{H} \otimes \mathcal{B} \otimes \mathcal{B}^* \otimes \mathcal{H}^*\,.
\end{tikzcd}$$

Alternatively, with Rem. 2.44 it is sufficient to observe that tensoring with an identity matrix $A \mapsto A \otimes I_{\mathcal{B}}$ is an algebra homomorphism.

Finally, it is immediate that the naturality squares (110) for a pointed endofunctor commute, by functoriality of the tensor product. □

Dually, we have:

Proposition 2.52 (Averaging quantum channels are QuantumStore transformations). *The averaging quantum channel* (46)

$$\begin{tikzcd}
(\text{-})\otimes(\mathcal{H}\otimes\mathcal{B})\otimes(\mathcal{H}\otimes\mathcal{B})^* \arrow[r, "\sim"] & (\text{-})\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[rr, "\mathrm{id}\otimes\mathrm{obt}^{\mathcal{B}^*\mathrm{Store}}_{\mathbb{1}}\otimes\mathrm{id}"] & & (\text{-})\otimes(\mathcal{H}\otimes\mathcal{H}^*) \\
|\text{-}\rangle\otimes|\psi,\beta\rangle\langle\beta',\psi'| \arrow[r, phantom, "="] & |\text{-}\rangle\otimes|\psi\rangle\otimes|\beta\rangle\langle\beta'|\otimes\langle\psi'| \arrow[rr, mapsto] & & |\text{-}\rangle\otimes|\psi\rangle\langle\beta'|\beta\rangle\langle\psi'|
\end{tikzcd}$$

is a comonadic QuantumState-transformation

$$\mathrm{Tr}^{\mathcal{B}} \;:\; (\mathcal{H}\otimes\mathcal{B})\mathrm{State} \xrightarrow{\;\mathrm{comon}\;} \mathcal{H}\mathrm{State}$$

and as such the component of a pointed endofunctor (110) *on* Mnd(QuType).

Proof. Since the structure maps of the $(\mathcal{H}\otimes\mathcal{B})$State-comonad are tensor products of structure maps of $\mathcal{H}$State and $\mathcal{B}$State, it is sufficient to show this for $\mathcal{H} = \mathbb{1}$, hence for the case that $\mathcal{H}$State = Id. But in this case $\mathrm{Tr}^{\mathcal{B}} = \mathrm{obt}^{\mathcal{B}^*\mathrm{Store}}_{(\text{-})}$, which we know to be a comonadic transformation according to (109).

Alternatively, it is immediate to explicitly check the required conditions. We have:

$$\begin{tikzcd}
(\text{-})\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[rrrr, "\mathrm{Tr}^{\mathcal{B}}_{(\text{-})}"] \arrow[dddd, "\mathrm{dupl}^{(\mathcal{H}\otimes\mathcal{B})^*\mathrm{Store}}_{(\text{-})}"'] & & & & (\text{-})\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[dddd, "\mathrm{dupl}^{\mathcal{H}^*\mathrm{Store}}_{(\text{-})}"] \\
& |\psi,\beta\rangle\langle\beta',\psi'| \arrow[rr, mapsto] \arrow[dd, mapsto] & & |\psi\rangle\langle\beta'|\beta\rangle\langle\psi'| \arrow[d, mapsto] & \\
& & & \langle\beta'|\beta\rangle \sum\limits_{w} |\psi\rangle\langle w|\otimes|w\rangle\langle\psi'| \arrow[d, equal] & \\
& \sum\limits_{w,b}|\psi,\beta\rangle\langle b,w|\otimes|w,b\rangle\langle\beta',\psi'| \arrow[rr, mapsto] & & \sum\limits_{w,b}|\psi\rangle\langle b|\beta\rangle\langle w|\otimes|w\rangle\langle\beta'|b\rangle\langle\beta'| & \\
(\text{-})\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[rrrr, "\mathrm{Tr}^{\mathcal{B}}_{(\cdots)}\circ(\mathrm{Tr}^{\mathcal{B}}_{(-)}\otimes\mathrm{id})"'] & & & & (\text{-})\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^*
\end{tikzcd}$$

and

$$\begin{tikzcd}
(\text{-})\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[rrrr, "\mathrm{Tr}^{\mathcal{B}}_{(-)}"] \arrow[dddd, "\mathrm{obt}^{(\mathcal{H}\otimes\mathcal{B})^*\mathrm{Store}}_{(\text{-})}"'] & & & & (\text{-})\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[dddd, "\mathrm{obt}^{\mathcal{H}^*\mathrm{Store}}_{(\text{-})}"] \\
& |\psi,\beta\rangle\langle\beta',\psi'| \arrow[rr, mapsto] \arrow[dd, mapsto] & & |\psi\rangle\langle\beta'|\beta\rangle\langle\psi'| \arrow[d, mapsto] & \\
& & & \langle\beta'|\beta\rangle\langle\psi'|\psi\rangle \arrow[d, equal] & \\
& \langle\beta',\psi'|\psi,\beta\rangle \arrow[rr, mapsto] & & \langle\beta',\psi'|\psi,\beta\rangle & \\
(\text{-}) \arrow[rrrr, equal] & & & & (\text{-})
\end{tikzcd}$$

Alternatively, with Rem. 2.44 it is sufficient to observe that partial tracing is a coalgebra homomorphism.

Finally, it is again immediate that the naturality squares (110) for a pointed endofunctor commute, by functoriality of the tensor product. □

Remark 2.53 (Partial trace)**.** On the other hand, partial trace is not a monadic QuantumState transformation beyond the trivial case of $\dim(\mathcal{B}) = 1$:

$$\begin{tikzcd}
(\text{-}) \arrow[rrrr, equal] \arrow[dddd, "\mathrm{ret}^{(\mathcal{H}\otimes\mathcal{B})\mathrm{State}}_{(\text{-})}"'] & & & & (\text{-}) \arrow[dddd, "\mathrm{ret}^{(\mathcal{H}\otimes\mathcal{B})\mathrm{State}}_{(\text{-})}"] \\
& |\text{-}\rangle \arrow[rr, mapsto] \arrow[dd, mapsto] & & |\text{-}\rangle \arrow[d, mapsto] & \\
& & & |\text{-}\rangle\otimes\sum\limits_{w}|w\rangle\langle w| \arrow[d, phantom, "\nparallel"] & \\
& |\text{-}\rangle\otimes\sum\limits_{w,b}|w,b\rangle\langle b,w| \arrow[rr, mapsto] & & |\text{-}\rangle\otimes\dim(\mathcal{B})\sum\limits_{w}|w\rangle\langle w| & \\
(\text{-})\otimes\mathcal{H}\otimes\mathcal{B}\otimes\mathcal{B}^*\otimes\mathcal{H}^* \arrow[rrrr, "(-)\otimes\mathrm{Tr}_{\mathcal{B}}"'] & & & & (\text{-})\otimes\mathcal{H}\otimes\mathcal{H}^*
\end{tikzcd}$$

Corollary 2.54 (Quantum states as transformations)**.** *Every unistochastic quantum channel* (56) *is a monadic QuantumState transformation (coupling and unitary evolution) followed by a comonadic QuantumState transformation (evolution and averaging).*

Interaction between QuantumState and QuantumEnvironment. Recall from §2.3 the monadic indefiniteness modality (QuantumReader) $\bigcirc_W$ and the comonadic randomness modality (QuantumCoreader) ☆_W.

Remark 2.55 (**QuantumEnvironment monad).** In its interaction with the QuantumState-monad, the epistemic modality $\bigcirc_W / \text{☆}_W$ or *W-Reader* (co)monad is suggestively referred to under its alternative name *W-environment* (co)-monad, and as such we will denote it "*W*Envm" and understand it as a Frobenius monad. Hence all the following names refer to the same monadic structure on linear types (cf. Prop. 2.35):

$$
\begin{array}{ccccc}
 & & \mathrm{Q}W\mathrm{Writer} & & \\
 & & \wr & & \\
\underset{W}{\bigcirc} & \simeq & W\mathrm{Envm} & \simeq & \underset{W}{\text{☆}} \\[6pt]
\mathrm{Monads} & \leftarrow & \mathrm{FrobMonad} & \rightarrow & \mathrm{Comonad}
\end{array}
$$

Proposition 2.56 (**QuantumState and QuantumEnvironment distribute).**
For $\mathcal{H} \,:\, \mathrm{QuType}^{\mathrm{fdm}}$ *and* $W \,:\, \mathrm{ClaType}^{\mathrm{fin}}$

(i) *the natural isomorphism*

$$
\begin{array}{ccccccc}
\mathcal{H}\mathrm{State}\big(\underset{W}{\bigcirc}\mathcal{K}\big) & \equiv & \big(\underset{W}{\bigoplus}\mathcal{K}\big)\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow[\sim]{\mathtt{distr}_{\mathcal{K}}^{\mathcal{H}\mathrm{State},\,\bigcirc_W}} & \underset{W}{\bigoplus}(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*) & \equiv & \underset{W}{\bigcirc}(\mathcal{H}\mathrm{State}(\mathcal{K})) \\
 & & (w,\,|\kappa\rangle)\otimes|\psi\rangle\langle\psi'| & \longleftrightarrow & (w,\,|\kappa\rangle\otimes|\psi\rangle\langle\psi'|) & & \\
\mathcal{H}^*\mathrm{Store}\big(\underset{W}{\text{☆}}\mathcal{K}\big) & \equiv & \big(\underset{W}{\bigoplus}\mathcal{K}\big)\otimes\mathcal{H}\otimes\mathcal{H}^* & \xleftarrow[\mathtt{distr}_{\mathcal{K}}^{\text{☆}_W,\,\mathcal{H}^*\mathrm{Store}}]{\sim} & \underset{W}{\bigoplus}(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*) & \equiv & \underset{W}{\text{☆}}(\mathcal{H}^*\mathrm{Store}(\mathcal{K}))
\end{array}
\tag{238}
$$

constitutes a distributivity transformation (113) *for*

- *the* $\mathcal{H}$State *monad over the W-indefiniteness monad,*
- *the W-randomness comonad distributing over the* $\mathcal{H}^*$Store*-comonad.*

(ii) *the same natural isomorphism, but understood as*

$$
\begin{array}{ccccccc}
\mathcal{H}^*\mathrm{Store}\big(\underset{W}{\bigcirc}\mathcal{K}\big) & \equiv & \big(\underset{W}{\bigoplus}\mathcal{K}\big)\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow[\sim]{\mathtt{distr}_{\mathcal{K}}^{\mathcal{H}^*\mathrm{Store},\,\bigcirc_W}} & \underset{W}{\bigoplus}(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*) & \equiv & \underset{W}{\bigcirc}(\mathcal{H}^*\mathrm{Store}(\mathcal{K})) \\
 & & (w,\,|\kappa\rangle)\otimes|\psi\rangle\langle\psi'| & \longleftrightarrow & (w,\,|\kappa\rangle\otimes|\psi\rangle\langle\psi'|) & & \\
\mathcal{H}\mathrm{State}\big(\underset{W}{\text{☆}}\mathcal{K}\big) & \equiv & \big(\underset{W}{\bigoplus}\mathcal{K}\big)\otimes\mathcal{H}\otimes\mathcal{H}^* & \xleftarrow[\mathtt{distr}_{\mathcal{K}}^{\text{☆}_W,\,\mathcal{H}\mathrm{State}}]{\sim} & \underset{W}{\bigoplus}(\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*) & \equiv & \underset{W}{\text{☆}}(\mathcal{H}\mathrm{State}(\mathcal{K}))
\end{array}
\tag{239}
$$

constitutes a distributivity transformation (124) *for*

- *the quantum* $\mathcal{H}^*$Store *comonad over the W-indefiniteness monad,*
- *the W-randomness comonad distributing over the* $\mathcal{H}$State*-monad.*

Proof. The required conditions (115) and (125) all hold rather immediately due to the ordinary distributivity of the tensor product (being a left adjoint) over the direct sum (being a coproduct, using here that W is a finite type).

□

Remark 2.57 (Distributivity is purely structural)**.** Since the distributivity laws in Prop. 2.56 are given just by the structure isomorphism of the underlying distributive monoidal category, we may and will leave it notationally implicit, writing $\underset{W}{\bigoplus}\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*$ as usual, without any parenthesis.

In generalization of Prop. 2.45, we have:

Proposition 2.58 (**Category of QuantumStore-context-dependent and Indefiniteness-effectful maps).** *For* $\mathcal{H} \,:\, \mathrm{QuType}^{\mathrm{fdm}}$ *and* $W \,:\, \mathrm{ClaType}^{\mathrm{fin}}$*, the jointly* $\mathcal{H}^*$Store*-contextful and* $\bigcirc_W$*-effectful morphisms* (123) *are in bijection with W-indexed sets of linear operators*

$$
\begin{array}{ccccc}
\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^* & \xrightarrow{\qquad\qquad o_{A_\bullet}\qquad\qquad} & \underset{W}{\bigcirc}\mathcal{K}' & & \\
|\kappa\rangle\otimes|\psi\rangle\langle\psi'| & \longmapsto & \big(w \,\mapsto\, \langle\psi', -|A_w|\kappa,\psi\rangle\big) & \longleftrightarrow & \big(A_w \,:\, \mathcal{K}\otimes\mathcal{H}\to\mathcal{K}'\otimes\mathcal{H}\big)_{w:W} \\
 & = & \big(w \,\mapsto\, \sum_{k'}\langle\psi', k'|A_w|\kappa,\psi\rangle\,|k'\rangle\big) & &
\end{array}
\tag{240}
$$

and their $\mathcal{H}^*\mathrm{Store}/\bigcirc_W$*-Kleisli composition* (126) *under the distributivity transformation* (2.56) *corresponds to the* W*-component wise operator products:*

$$(\mathtt{bind}^{\bigcirc_W}_{\mathcal{K}''}\mathcal{O}_{B_\bullet}) \;\circ\; \mathtt{distr}^{\mathcal{H}^*\mathrm{Store},\,\bigcirc_W}_{\mathcal{K};} \;\circ\; (\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}}\mathcal{O}_{A_\bullet}) \quad = \quad \mathcal{O}_{(B\cdot A)_\bullet}\,.$$

Proof. By the general formula (126) and with Rem. 2.57:

$$\begin{tikzcd}
& \mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\mathcal{O}_{A_\bullet}\otimes\mathcal{H}\otimes\mathcal{H}^*"] & \bigoplus\limits_W\mathcal{K}'\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\oplus_W\,\mathcal{O}_{B_\bullet}"] \arrow[drr, "\mathtt{bind}^{\bigcirc_W}\mathcal{O}_{B_\bullet}"'] & \bigoplus\limits_W\bigoplus\limits_W\mathcal{K}'' \arrow[dr, "\mathrm{join}^{\bigcirc_W}_{\mathcal{K}}"] & \\
\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[ur, "\mathrm{dupl}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}}"] \arrow[urr, "\mathtt{extend}^{\mathcal{H}^*\mathrm{Store}}_{\mathcal{K}}\mathcal{O}_{A_\bullet}"'] & & & & \bigoplus\limits_W\mathcal{K}'' \\
& |\kappa\rangle\otimes\sum\limits_h|\psi\rangle\langle h|\otimes|h\rangle\langle\psi'| \arrow[r, mapsto] & \Big(w\mapsto\sum\limits_{h,k'}\langle h,k'|A_w|\kappa,\psi\rangle\,|k'\rangle\otimes|h\rangle\langle\psi'|\Big) \arrow[dr, mapsto] & & \\
|\kappa\rangle\otimes|\psi\rangle\langle\psi'| \arrow[ur, mapsto] & & & \Big(w\mapsto\underbrace{\sum\limits_{h,k'}\langle\psi',-|B_w|k',h\rangle\langle h,k'|A_w|\kappa,\psi\rangle}_{\langle\psi',-|B_w\cdot A_w|\kappa,\psi\rangle}\Big) &
\end{tikzcd}$$

□

Example 2.59 (State preparation with Probability weights). Given $W\;:\;\mathrm{ClaType}^{\mathrm{fin}}$ we have the following basic examples of W-environment-contextful and $\mathrm{Q}W$-effective maps:

(i) The map prep which at environmental parameter $w : W$ produces ("prepares") the corresponding pure basis state $|h\rangle\langle h|$;

(ii) for $p\;:\;W\to\mathbb{R}_{\geq 0}$ a (probability) measure, the map $\mathrm{weigh}_{p_\bullet}$ which at environmental parameter $w\;:\;W$ produces the identity (density) matrix with coefficient p_w.

$$\begin{tikzcd}
\mathrm{prep}\;:\;W\mathrm{Envm}(\mathbb{1}) \arrow[r] & \mathrm{Q}W\mathrm{State}(\mathbb{1}) \\
\underset{W}{\star}\,\mathbb{C} \arrow[r] & \mathrm{Q}W\otimes\mathrm{Q}W^* \\
(w,1) \arrow[r, mapsto] & |h\rangle\langle h|
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathrm{weigh}_{p_\bullet}\;:\;W\mathrm{Envm}(\mathbb{1}) \arrow[rr] & & \mathrm{Q}W\mathrm{State}(\mathbb{1}) \\
\underset{W}{\star}\,\mathbb{C} \arrow[r, "p_\bullet"] & \mathbb{C} \arrow[r, "\mathrm{ret}^{\mathcal{H}\mathrm{State}}_{\mathbb{1}}"] & \mathrm{Q}W\otimes\mathrm{Q}W^* \\
(w,1) \arrow[r, mapsto] & p_w \arrow[r, mapsto] & p_w\cdot\sum\limits_h|h\rangle\langle h|\,.
\end{tikzcd}$$

Their two-sided Kleisli composition prepares the mixed state in which the pure state $|h\rangle$ appears with weight p_w:

$$\begin{tikzcd}
\underset{W}{\star}\,\mathbb{C} \arrow[r, "\mathrm{dupl}^{\star_W}_{\mathbb{1}}"] \arrow[rrrr, bend left, "\mathrm{weight}_{p_\bullet}\;>=>\;\mathrm{prep}"] & \underset{W}{\star}\,\underset{W}{\star}\,\mathbb{C} \arrow[r, "\star_W\mathrm{weigh}_{p_\bullet}"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\mathrm{prep}\otimes\mathcal{H}\otimes\mathcal{H}^*"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\mathrm{join}^{\mathcal{H}\mathrm{State}}_{\mathbb{1}}"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^* \\
(w,1) \arrow[r, mapsto] & (w,(w,1)) \arrow[r, mapsto] & (w,p_w\,I_{\mathcal{H}}) \arrow[r, mapsto] & p_w\,I_{\mathcal{H}}\otimes|w\rangle\langle w| \arrow[r, mapsto] & p_w\,|w\rangle\langle w|
\end{tikzcd}$$

$$\begin{tikzcd}
\underset{W}{\star}\,\mathbb{C} \arrow[r, "\mathrm{dupl}^{\star_W}_{\mathbb{1}}"] \arrow[rrrr, bend left, "\mathrm{prep}\;>=>\;\mathrm{weigh}_{p_\bullet}"] & \underset{W}{\star}\,\underset{W}{\star}\,\mathbb{C} \arrow[r, "\star_W\,\mathrm{prep}"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\mathrm{weigh}_{p_\bullet}\otimes\mathcal{H}\otimes\mathcal{H}^*"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^* \arrow[r, "\mathrm{join}^{\mathcal{H}\mathrm{State}}_{\mathbb{1}}"] & \underset{W}{\star}\,\mathcal{H}\otimes\mathcal{H}^* \\
(w,1) \arrow[r, mapsto] & (w,(w,1)) \arrow[r, mapsto] & (w,\,|w\rangle\langle w|) \arrow[r, mapsto] & p_w\,|w\rangle\langle w|\otimes I_{\mathrm{Q}W} \arrow[r, mapsto] & p_w\,|w\rangle\langle w|
\end{tikzcd}$$

Lemma 2.60 (Distributive monad transformations act on context/effectful-maps). *For* C *a comonad distributing* (124) *over a pair of monads* $\mathcal{E}$, $\mathcal{E}'$

$$\mathtt{distr}^{C,\mathcal{E}} : C\circ\mathcal{E}\longrightarrow\mathcal{E}\circ C\,,\qquad\qquad \mathtt{distr}^{C,\mathcal{E}'} : C\circ\mathcal{E}'\longrightarrow\mathcal{E}'\circ C$$

then a monad transformation (103)

$$\mathtt{trans}^{\mathcal{E}\longrightarrow\mathcal{E}'} : \mathcal{E}\to\mathcal{E}'$$

which is compatible with the two distributive laws in that it makes the following diagram commute

$$\begin{tikzcd}
C(\mathcal{E}(-)) \arrow[r, "C(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{(-)})"] \arrow[d, "\mathtt{distr}^{C,\mathcal{E}}_{(-)}"'] & C(\mathcal{E}'(-)) \arrow[d, "\mathtt{distr}^{C,\mathcal{E}'}_{(-)}"] \\
\mathcal{E}(C(-)) \arrow[r, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{C(-)}"'] & \mathcal{E}'(C(-))\,,
\end{tikzcd} \tag{241}$$

respects the Kleisli composition of context-effectful maps (126) *just as it does respect* (105) *the plain Kleisli composition* (69) *of purely-effectful maps:*

$$
\left.\begin{aligned}
\mathrm{prog}_{12} &: C(D_1) \to \mathcal{E}(D_2) \\
\mathrm{prog}_{23} &: C(D_2) \to \mathcal{E}(D_3)
\end{aligned}\right\}
\quad \vdash \quad
\begin{aligned}
& \big(\mathtt{trans}^{\mathcal{E}\,\to\,\mathcal{E}'}_{D_2} \circ \mathrm{prog}_{12}\big) \;\texttt{>=>}\; \big(\mathtt{trans}^{\mathcal{E}\,\to\,\mathcal{E}'}_{D_3} \circ \mathrm{prog}_{23}\big) \\
= \quad & \mathtt{trans}^{\mathcal{E}\,\to\,\mathcal{E}'}_{D_2} \circ \big(\mathrm{prog}_{12} \;\texttt{>=>}\; \mathrm{prog}_{23}\big)\,.
\end{aligned}
\tag{242}
$$

Proof. Consider the following diagram:

$$\begin{tikzcd}
C(D_1) \arrow[r, "\mathrm{dupl}^{C}_{D_1}"] \arrow[rr, bend left, "\mathtt{extend}^{C}\mathrm{prog}_{12}"] \arrow[drr, bend right, "\mathtt{extend}^{C}\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ\mathrm{prog}_{12}\big)"'] & C(C(D_1)) \arrow[r, "C(\mathrm{prog}_{12})"] \arrow[dr, "C\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ\mathrm{prog}_{12}\big)"'] & C(\mathcal{E}(D_2)) \arrow[r, "\mathtt{distr}^{C,\mathcal{E}}_{D_2}"] \arrow[d, "C\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\big)"] & \mathcal{E}(C(D_2)) \arrow[r, "\mathcal{E}(\mathrm{prog}_{23})"] \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{C(D_2)}"] \arrow[drr, bend left, "\mathtt{bind}^{\mathcal{E}}\mathrm{prog}_{23}"] & \mathcal{E}(\mathcal{E}(D_3)) \arrow[d, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{\mathcal{E}(D_3)}"] \arrow[dr, "\mathrm{join}^{\mathcal{E}}_{D_3}"] & \\
 & & C(\mathcal{E}'(D_3)) \arrow[r, "\mathtt{distr}^{C,\mathcal{E}'}_{D_2}"] & \mathcal{E}'(C(D_2)) \arrow[r, "\mathcal{E}'(\mathrm{prog}_{23})"] \arrow[dr, "\mathcal{E}'\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ\mathrm{prog}_{23}\big)"'] \arrow[ddrr, bend right, "\mathtt{bind}^{\mathcal{E}'}\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_2}\circ\mathrm{prog}_{23}\big)"'] & \mathcal{E}'(\mathcal{E}(D_3)) \arrow[d, "\mathcal{E}'\big(\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_3}\big)"] & \mathcal{E}(D_3) \arrow[dd, "\mathtt{trans}^{\mathcal{E}\to\mathcal{E}'}_{D_3}"] \\
 & & & & \mathcal{E}'(\mathcal{E}'(D_3)) \arrow[dr, "\mathrm{join}^{\mathcal{E}'}_{D_3}"] & \\
 & & & & & \mathcal{E}'(D_3)\,.
\end{tikzcd}$$

Here the middle square commutes by the distributivity assumption (241), the square to the right of it due to naturality of the transformation $\mathtt{trans}^{\mathcal{E}\,\to\,\mathcal{E}'}$ and the far right square due to its monad transformation property (108). Therefore the total diagram commutes. But its total top and right composite morphism is the right-hand side of (242), while its total bottom left (diagonal) composite morphism is the left-hand side of (242), thus proving their equality. □

It follows immediately that:

Lemma 2.61 (QuantumState transformations compatible with distributivity over Quantum Reader). *Every transformation* (103) *between quantum state monads* (223) *is compatible* (241) *with the canonical distributivity* (239) *over the QuantumReader monads.*

Proof. Use Lem. 2.64. □

As a corollary of Lem. 2.60 and Lem. 2.61:

Proposition 2.62 (Preserving Quantum Kleisli composition). *Given a quantum channel which acts as a QuantumState transformation* (such as unitary channels by Prop. 2.48 and uniform coupling channels by Prop. 2.51)

$$
\mathrm{chan} : \mathcal{H}\mathrm{State} \to \mathcal{H}'\mathrm{State}
$$

then composition of this channel with maps that are Randomness-contextful and QuantumState-effectful preserves their Kleisli composition (126)*, in that:*

$$
\left.\begin{aligned}
\mathrm{prog}_{12} &: \underset{W}{\text{☆}}\mathcal{K} \to \mathcal{K}' \otimes \mathcal{H} \otimes \mathcal{H}^* \\
\mathrm{prog}_{23} &: \underset{W}{\text{☆}}\mathcal{K}' \to \mathcal{K}'' \otimes \mathcal{H} \otimes \mathcal{H}^*
\end{aligned}\right\}
\quad \vdash \quad
\begin{aligned}
& \big(\mathrm{chan}_{\mathcal{K}'} \circ \mathrm{prog}_{12}\big) \;\texttt{>=>}\; \big(\mathrm{chan}_{\mathcal{K}''} \circ \mathrm{prog}_{23}\big) \\
= \quad & \mathrm{chan}_{\mathcal{K}''} \circ \big(\mathrm{prog}_{12} \;\texttt{>=>}\; \mathrm{prog}_{23}\big)\,.
\end{aligned}
\tag{243}
$$

Indefinite QuantumStates. We may now combine the indefiniteness-effects which model quantum measurement and classical control (§2.4) with the QuantumState-effects that model mixed states:

Definition 2.63 (Category of Quantum State Effects). We write

$$\mathrm{QuEffect} \;\equiv\; \mathrm{Mnd}(\mathrm{QuType})$$

for the category of monads – with monad transformations (103) between them – on the category of quantum types. And we write

$$\mathrm{QuStateEffect} \hookrightarrow \mathrm{QuEffect} \tag{244}$$

for its full subcategory on the QuantumState monads $\mathcal{H}\mathrm{State}$ for $\mathcal{H} \,:\, \mathrm{QuType}^{\mathrm{fdm}}$.

Lemma 2.64 (Natural transformations between tensoring functors). *For* $\mathcal{V}_1, \mathcal{V}_2 \,:\, \mathrm{QuType}^{\mathrm{fdm}}$*, with*

$$(-) \otimes \mathcal{V}_i \,:\, \mathrm{QuType} \to \mathrm{QuType}$$

the functors of tensoring with these objects, then all natural transformations between them

$$f_{(-)} \,:\, (-) \otimes \mathcal{V}_1 \to (-) \otimes \mathcal{V}_1$$

are given by tensoring with the linear map that is their value on the tensor unit:

$$f_{\mathcal{K}} \;\simeq\; \mathcal{K} \otimes f_{\mathbb{1}}\,.$$

Proof. This follows by the QuType-enriched Yoneda lemma after observing that the tensor functors $(-) \otimes \mathcal{V}_i$ are representable

$$(-) \otimes \mathcal{V}_i \;\simeq\; (-) \otimes (\mathcal{V}_i^*)^* \;\simeq\; \mathcal{V}_i^* \multimap (-)\,.$$

□

Lemma 2.65 (QuantumState transformations are algebra homomorphisms). *QuantumState transformations are in natural bijection to monoid homomorphisms*

$$\begin{tikzcd} \mathrm{QuantumStateEffects} \arrow[r, hook] & \mathrm{Mon}(\mathrm{QuType}) \\ \mathcal{H}\mathrm{State} \arrow[r, mapsto] & \mathcal{H} \otimes \mathcal{H}^* \end{tikzcd}$$

Proof. Via Rem. 2.44, it is clear that natural transformations of tensor form

$$\begin{tikzcd} \mathcal{H}_1\mathrm{State} \arrow[r, "(-)\otimes\phi"] & \mathcal{H}_2\mathrm{State} \\ (-) \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "\mathrm{id}\otimes\phi"] & (-) \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^* \end{tikzcd}$$

are monad transformations if and only if ϕ is an algebra homomorphism. Therefore, it only remains to observe that all natural transformations are necessarily of this tensor form, which is the statement of Lem. 2.64. □

As a corollary:

Lemma 2.66 (QuantumState and linear maps). *The isomorphisms of QuantumState effects are given by conjugation with invertible linear maps. In particular, a natural transformation of the form*

$$\begin{tikzcd}
\mathrm{chan}^{H} \;:\; \mathcal{H}_1\mathrm{State} \arrow[r] & \mathcal{H}_2\mathrm{State} \\
(-) \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "(-)\otimes(U\otimes U^{\dagger *})"] & (-) \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

is a QuantumState-transformation if and only if $U \;:\; \mathcal{H}_1 \to \mathcal{H}_2$ is unitary.

Definition 2.67 (IndefiniteQuantumState-monad). For $W \;:\; \mathrm{ClaType}^{\mathrm{fin}}$ and $\mathcal{H} \;:\; \mathrm{QuType}^{\mathrm{fdm}}$, we have the composite monad (112) of the QuantumState- with the Indefiniteness-monad:

$$\begin{tikzcd}
\underset{W}{\bigcirc} \circ \mathcal{H}\mathrm{State} \;:\; \mathrm{QuType} \arrow[r] & \mathrm{QuType} \\
\mathcal{K} \arrow[r, mapsto] & \underset{W}{\oplus} \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^*
\end{tikzcd}$$

Proposition 2.68 (IndefiniteQuantumState-effectful transformations). *The monad transformations* (103) *from a QuantumState-monad (Def. 2.43) to an IndefiniteQuantumState-monad (Def. 2.67)*

$$f \;:\; \mathcal{H}_1\mathrm{State} \longrightarrow \bigcirc_W \circ \mathcal{H}_2\mathrm{State}$$

are in natural bijection to W-tuples of algebra homomorphisms.

Proof. By Lem. 2.64, the underlying natural transformation is given by tensoring

$$f_{\mathcal{K}} \;\simeq\; \mathcal{K} \otimes f_{\mathbb{1}}$$

with a linear map

$$\begin{tikzcd}
f_{\mathbb{1}} \;:\; \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r] & \underset{W}{\oplus} \mathcal{H}_1 \otimes \mathcal{H}_1^* \\
A \arrow[r, mapsto] & \underset{w}{\oplus} f_{\mathbb{1}}(A)_w \,.
\end{tikzcd}$$

In terms of this, the monad-transformation property of $f_{(-)}$

$$\begin{tikzcd}
\mathcal{K} \arrow[r, equals] \arrow[d, "\mathrm{ret}^{\mathcal{H}\mathrm{State}}_{\mathcal{K}}"'] & \mathcal{K} \arrow[d, "\mathrm{ret}^{\bigcirc_W \circ \mathcal{H}\mathrm{State}}_{\mathcal{K}}"] \\
\mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "f_{\mathcal{K}}"'] & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[r, "f_{\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*}"] \arrow[d, "\mathrm{join}^{\mathcal{H}_1\mathrm{State}}_{\mathcal{K}}"'] & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^* \arrow[r, "\underset{W}{\bigcirc} f_{\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*}"] & \underset{W}{\bigcirc}\underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^* \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^* \arrow[d, "\mathrm{join}^{\bigcirc_W \circ \mathcal{H}_2\mathrm{State}}_{\mathcal{K}}"] \\
\mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[rr, "f_{\mathcal{K}}"'] & & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

translates to the condition for W-indexed monoid homomorphisms, as claimed:

$$\begin{tikzcd}
\mathbb{1} \arrow[rrr, equals] \arrow[ddd] & & & \mathbb{1} \arrow[ddd] \\
& 1 \arrow[r, mapsto] \arrow[d, mapsto] & 1 \arrow[d, mapsto] & \\
& 1_1 \arrow[r, mapsto] & \underset{w}{\oplus} 1_2 & \\
\mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[rrr, "f_{\mathbb{1}}"'] & & & \underset{W}{\bigcirc} \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{H}_1 \otimes \mathcal{H}_1 \otimes \mathcal{H}_1 \otimes \mathcal{H}_1 \arrow[rrr, "f_{\mathbb{1}} \otimes f_{\mathbb{1}}"] \arrow[ddd, "(-)\cdot(-)"'] & & & \big(\underset{W}{\bigoplus} \mathcal{H}_2 \otimes \mathcal{H}_2\big) \otimes \big(\underset{W}{\bigoplus} \mathcal{H}_2 \otimes \mathcal{H}_2\big) \arrow[ddd, "\underset{w}{\oplus}\big((-)_w \cdot (-)_w\big)"] \\
& A \otimes B \arrow[r, mapsto] \arrow[d, mapsto] & \big(\underset{w}{\oplus}\, f_{\mathbb{1}}(A)_w\big) \otimes \big(\underset{w}{\oplus}\, f_{\mathbb{1}}(B)_w\big) \arrow[d, mapsto] & \\
& A \cdot B \arrow[r, mapsto] & \underset{w}{\oplus} f_{\mathbb{1}}(A \cdot B)_w & \\
\mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[rrr, "f_{\mathbb{1}}"'] & & & \underset{W}{\bigcirc} \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

□

Proposition 2.69 (Indefiniteness-effect on QuantumState-effects). *For* W : ClaType$^{\text{fin}}$ *and* $\mathcal{H}$: QuType$^{\text{fdm}}$*, the construction of the IndefiniteQuantumState-monad* $\underset{W}{\bigcirc} \circ \mathcal{H}\text{State}$ *(Def. 2.67) extends to a relative monad* (101) *on, in turn, the category of quantum state effects (Def. 2.63):*

$$\begin{tikzcd}
\bigcirc_W \circ (-) \;:\; \text{QuStateEffect} \arrow[r] & \text{QuEffect} \\
\mathcal{H}\text{State} \arrow[r, mapsto] & \underset{W}{\bigcirc} \circ \mathcal{H}\text{State}
\end{tikzcd}$$

relative to the full inclusion (244).

Proof. The return-operation is

$$\begin{array}{lcl}
\mathrm{ret}^{\bigcirc_W \circ}_{\mathcal{H}\text{State}} & : & \mathcal{H}\text{State} \longrightarrow \underset{W}{\bigcirc} \circ \mathcal{H}\text{State} \\
(\mathrm{ret}^{\bigcirc_W \circ}_{\mathcal{H}\text{State}})_{\mathcal{K}} & : & \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \xrightarrow{\mathrm{ret}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*}} \underset{W}{\bigoplus} \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \\
 & & |\kappa\rangle \otimes |\psi\rangle\langle\psi'| \longmapsto \sum_w (w,\, |\kappa\rangle \otimes |\psi\rangle\langle\psi'|)
\end{array} \tag{245}$$

and the bind-operation takes a monad transformation $f : \mathcal{H}_1\text{State} \to \underset{W}{\bigcirc} \circ \mathcal{H}_2\text{State}$ to $\text{join}^{\bigcirc_W} \circ \underset{W}{\bigcirc} f$. That this satisfies the axioms of a relative monad follows immediately from the monad structure on $\bigcirc_W$: QuType → QuType. But for this to be well-defined as a monad on monads, we do in addition need to check that the return- and bind-operations now are actually morphisms in QuEffect:

That (245) is a monad transformation follows by the definition (114) of the composite monad alone, which immediately shows that these diagrams commute:

$$\begin{tikzcd}
\mathcal{K} \arrow[r, equal] \arrow[d, "\mathrm{ret}^{\mathcal{H}\text{State}}_{\mathcal{K}}"'] & \mathcal{K} \arrow[d, "\mathrm{ret}^{\bigcirc_W \circ \mathcal{H}\text{State}}_{\mathcal{K}}"] \\
\mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*}"] & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^*
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[r, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^*}"] \arrow[d, "\mathrm{join}^{\mathcal{H}\text{State}}_{\mathcal{K}}"'] & \underset{W}{\bigcirc}(\mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^*) \arrow[r, "\underset{W}{\bigcirc} \mathrm{ret}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*} \otimes \mathcal{H} \otimes \mathcal{H}^*"] \arrow[dr, "\underset{W}{\bigcirc} \mathrm{join}^{\mathcal{H}\text{State}}_{\mathcal{K}}" description] & \underset{W}{\bigcirc}\Big(\big(\underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^*\big) \otimes \mathcal{H} \otimes \mathcal{H}^*\Big) \arrow[d, "\mathrm{join}^{\bigcirc_W \circ \mathcal{H}\text{State}}_{\mathcal{K}}"] \\
\mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \arrow[rr, "\mathrm{ret}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*}"] & & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H} \otimes \mathcal{H}^* \otimes \mathcal{H} \otimes \mathcal{H}^*
\end{tikzcd}$$

That the effect-binding of f is still a monad transformation follows from the fact that f itself is assumed to be a monad transformation and using Prop. 2.68:

$$\begin{tikzcd}
\mathcal{K} \arrow[rrr, equal] \arrow[ddd, "\mathrm{ret}^{\mathcal{H}_1\text{State}}_{\mathcal{K}}"'] & & & \mathcal{K} \arrow[ddd, "\mathrm{ret}^{\bigcirc_W \circ \mathcal{H}_1\text{State}}_{\mathcal{K}}"] \\
& |\kappa\rangle \arrow[r, mapsto] \arrow[d, mapsto] & |\kappa\rangle \arrow[d, mapsto] & \\
& |\kappa\rangle \otimes 1_1 \arrow[r, mapsto] & |\kappa\rangle \otimes \underset{w}{\oplus} 1_2 & \\
\mathcal{K} \otimes \mathcal{H}_1 \otimes \mathcal{H}_1^* \arrow[rrr, "f_{\mathcal{K}}"'] & & & \underset{W}{\bigcirc} \mathcal{K} \otimes \mathcal{H}_2 \otimes \mathcal{H}_2^*
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=small]
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[rrrr, "\mathrm{join}_{\mathcal{K}}^{\bigcirc_W\circ\mathcal{H}_1\mathrm{State}}"] \arrow[dd, "\underset{W}{\bigcirc} f_{\bigcirc_W\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*}"] & & & & \underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[ddd, "\underset{W}{\bigcirc} f_{\mathcal{K}}"] \\
& {|\kappa\rangle\otimes(w,\,A)\otimes(w',\,A')} \arrow[r, mapsto] \arrow[dd, mapsto] & {} & {|\kappa\rangle\otimes(w,\,\delta_w^{w'}A\cdot A')} \arrow[dddd, mapsto] & \\
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[d, "\mathrm{join}^{\bigcirc_W}_{\bigcirc_W\mathcal{K}\otimes\mathcal{H}\otimes\mathcal{H}^*\otimes\mathcal{H}\otimes\mathcal{H}^*}"] & & & & \\
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_1\otimes\mathcal{H}_1^*\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[d, "\underset{W}{\bigcirc}\underset{W}{\bigcirc}f_{\mathcal{K}}\otimes\mathcal{H}_2\otimes\mathcal{H}_2"] & {|\kappa\rangle\otimes(w,\,A)\otimes(w',\,f_{\mathbb{1}}(A')_{w'})} \arrow[dd, mapsto] & & & \underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[ddd, "\mathrm{join}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*}"] \\
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[dd, "\bigcirc_W\mathrm{join}^{\bigcirc_W}_{\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*"] & & & & \\
& {|\kappa\rangle\otimes(w,\,f_{\mathbb{1}}(A)_w)\otimes(w',\,f_{\mathbb{1}}(A')_{w'})} \arrow[r, mapsto] & {} & {|\kappa\rangle\otimes(w,\,\delta_w^{w'}f_{\mathbb{1}}(A\cdot A')_w)} & \\
\underset{W}{\bigcirc}\underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*\otimes\mathcal{H}_2\otimes\mathcal{H}_2^* \arrow[rrrr, "\mathrm{join}_{\mathcal{K}}^{\bigcirc_W\circ\mathcal{H}_1\mathrm{State}}"'] & & & & \underset{W}{\bigcirc}\mathcal{K}\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*
\end{tikzcd}$$

□

Proposition 2.70 (Enhancing parameterized quantum circuits to parameterized quantum channels). *The W-componentwise unitary $\bigcirc_W$-effectful maps of* QuType *lift via* (213) *to* $\bigcirc_W\circ$*-effectful maps on* QuEffect

$$\begin{tikzcd}
\mathrm{QuType}^{\mathrm{untr}}_{\bigcirc_W} \arrow[r] & \mathrm{QuEffect}_{\bigcirc_W\circ}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{H}_1 \arrow[r, mapsto] \arrow[dd, "U_\bullet"] & \mathcal{H}_1\mathrm{State} \arrow[dd, "\mathrm{chan}^{U_\bullet}"] & (-)\otimes\mathcal{H}_1\otimes\mathcal{H}_1^* \arrow[d, "(-)\otimes U_\bullet\otimes U_\bullet^{\dagger *}"] \\
& & (-)\otimes\underset{W}{\bigoplus}\mathcal{H}_1\otimes\underset{W}{\bigoplus}\mathcal{H}_1^* \arrow[d, "(-)\otimes\mathrm{pair}^{\bigcirc_W}_{\mathcal{H}_2,\mathcal{H}_2^*}"] \\
\underset{W}{\bigcirc}\mathcal{H}_2 & \underset{W}{\bigcirc}\circ\mathcal{H}_2\mathrm{State} & \underset{W}{\bigoplus}(-)\otimes\mathcal{H}_2\otimes\mathcal{H}_2^*
\end{tikzcd}$$

Proof. It remains to see that the paired Kleisli maps (213) are indeed QuantumState monad-transformations. This is ensured by the unitarity assumption, as in Prop. 2.48. □

3 Quantum Language

With all quantum effects identified – in §2 – as (co)monads definable through the Motivic Yoga (Def. 2.18), we may follow established language paradigms for monadic effects (Lit. 1.19) to obtain a natural quantum language – to be called `QS` [34] – that should be be embeddable as a domain-specific language (Lit. 1.17) into any dependent linear type theory which verifies the Motivic Yoga, notably into `LHoTT` (Lit. 1.8).

3.1 Pseudocode Design

In the spirit of traditional `do`-notation for monadic computational effects (Lit. 1.19) our ambition is to find (sugaring to) an accurate but neatly intelligible formal language for the monadic quantum effects which is close to a natural description of the coded processes. For that purpose, we employ syntactic sugar both for effect binding and for pure effects (67):

(i) Syntactic sugar for effect-binding.
For effect-binding we use traditional `do`-notation but in the more verbose form of `for`...`do`-blocks (132),

(ii) Syntactic sugar for pure effects.
We furthermore sugar the `return`-operation of each effect such as to notationally indicate the nature of the pure datum that is being returned (252).

First we discuss the declaration of plain linear maps (quantum gates). Recall our convention (162) to write an "open colon" " $\mathrel{\substack{\circ\\ \circ}}$ " for typing judgements in the context of the linear tensor unit, which we will use throughout.

Declaration of linear maps out of the tensor unit. To start with, in declaring linear maps out of the linear tensor unit it should, by linearity, be sufficient to declare the value on the unit element

$$\begin{aligned} \phi &\mathrel{\substack{\circ\\ \circ}} \ \mathbb{1} \multimap \mathcal{H} \\ \phi &\equiv 1 \mapsto \phi(1)\,. \end{aligned} \tag{246}$$

Self-evident as this may seem, this is ultimately a consistency demand on the ambient linear type theory, which must provide the corresponding elimination rule for the tensor unit. In `LHoTT` this is the case: [Ri22a, p. 55] speaks of the $\mathbb{S}$-*elimination*- or $\mathbb{S}$-*induction*-rule (where the notation "$\mathbb{S}$" alludes to the sphere spectrum, which is the tensor unit in the expected model of `LHoTT` in parameteroized plain spectra, aka $\mathbb{S}$-modules.)

Declaration of linear maps out of a linear span. Recall that the quantization modality Q (Def. 2.13) is just the quantumly-modality $\triangleright$ restricted to classical types along the operation $\mathbb{1} \times (-)$

$$\mathrm{Q} \;\equiv\; \triangleright\big((-) \times \mathbb{1}\big)\,.$$

Regarded as a restriction of $\triangleright$, it binds not just Q-effects but generally $\triangleright$-effects, cf. (164). Now, $\triangleright$ is idempotent (155), meaning that for every linear type is a free $\triangleright$-modale: $\mathcal{H} = \triangleright\mathcal{H}$.

In conclusion this means that do-notation applies to to declare linear maps (quantum gates) of the form $G \mathrel{\substack{\circ\\ \circ}} \mathrm{Q}W_1 \multimap \mathcal{H}$, whose domain is equipped with a linear basis W with corresponding basis vectors are denoted $|w\rangle \mathrel{\substack{\circ\\ \circ}} \mathrm{Q}W$ (165), while the codomain may be any linear type.

In natural language, we would describe such a map by declaring what it does *for* a given basis vector $|w\rangle$ – namely sending it to $|G_w\rangle := G|w\rangle$ – and we want this natural description to essentially already be our syntax, as follows:

$$\begin{aligned} G &\mathrel{\substack{\circ\\ \circ}} \ \mathrm{Q}W \multimap \mathcal{H} \\ G &\equiv \left[\begin{array}{l} \texttt{for}\ \ |w\rangle \\ \ \texttt{do}\ \ G|w\rangle\,. \end{array}\right. \end{aligned} \tag{247}$$

Indeed, this is the traditional `do`-notation (Lit. 1.19) in `for`...`do`-form (132), applied to the quantum modality, except for a further sugaring of the plain "w" to its pure-effect incarnation "$|w\rangle$". This notation naturally reflects that QW is freely *generated*

---

[34] 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 spherical homotopy types S^n to the "sphere spectrum" traditionally denoted "QS^n".

(i) in the sense of generating sets of vector spaces: by the vectors $|w\rangle$

(ii) in the sense of free $\triangleright$-modales: by the elements $(w, 1) : W \times \mathbb{1}$,

and the operation which relates these two incarnations of the generators is $\mathtt{return}^{\mathrm{Q}}_{W}$ (165), namely:

$$|w\rangle \;\;\equiv\;\; \mathtt{return}^{\mathrm{Q}}_{W}(w) \;\;\equiv\;\; \mathtt{return}^{\triangleright}_{W\times\mathbb{1}}((w,1)) \;:\; \mathrm{Q}W\,. \tag{248}$$

Therefore the natural do-notation for the $\triangleright$-bind operation on a linear map

$$G|{-}\rangle \;:\; W \times \mathbb{1} \multimap \mathcal{H}\,,$$

– which according to (246) is specified by its value on the elements $(w, 1)$ whose natural name in $\mathrm{Q}W$ is $|w\rangle$ – is the above (247).

Declaration of linear maps out of a tensor product. In the same vein, for declaring a linear map out of a tensor product, one would naturally want the following syntax, defining its value *for* each decomposable tensor:

$$\begin{aligned} G \;&:\; \mathrm{Q}W_1 \otimes \mathrm{Q}W_2 \multimap \mathcal{H} \\ G \;&\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |w_1\rangle \otimes |w_2\rangle \\ \quad \mathtt{do}\;\; G(|w_1\rangle \otimes |w_2\rangle) \end{array}\right. \end{aligned} \tag{249}$$

Now understanding

$$\mathrm{Q}W_1 \otimes \mathrm{Q}W_2 \;=\; \triangleright(\mathrm{Q}W_1 \otimes \mathrm{Q}W_2)$$

again as a restriction of the quantum modality – to the entanglement relative monad (167) – we may indeed take this as the coresponding do-notation subject only to the further convention that, as before, we refer to the argument via its pure effect incarnation:

$$|w_1\rangle \otimes |w_2\rangle \;\;\equiv\;\; \mathtt{return}^{\mathrm{Q}(\text{-})\otimes\mathrm{Q}(\text{-})}_{(W_1,\,W_2)}(w_1, w_2)\,.$$

For example, with (247) and (249) the operations which witness the strong $\otimes$-monoidal property of Q may thus be coded as follows:

$$\begin{aligned} \mu \;&:\; \mathrm{Q}W_1 \otimes \mathrm{Q}W_2 \multimap \mathrm{Q}(W_1 \times W_2) & \mu^{-1} \;&:\; \mathrm{Q}(W_1 \times W_2) \multimap \mathrm{Q}W_1 \otimes \mathrm{Q}W_2 \\ \mu \;&\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |w_1\rangle \otimes |w_2\rangle \\ \quad \mathtt{do}\;\; |w_1, w_2\rangle \end{array}\right. & \mu^{-1} \;&\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |w_1, w_2\rangle \\ \quad \mathtt{do}\;\; |w_1\rangle \otimes |w_2\rangle \end{array}\right. \end{aligned} \tag{250}$$

and the tensor product on maps out of linear spans is given by

$$\begin{array}{l} G : \mathrm{Q}W \multimap \mathcal{H} \\ G' : \mathrm{Q}W' \multimap \mathcal{H}' \end{array} \quad\vdash\quad \begin{aligned} G \otimes G' \;&:\; \mathrm{Q}W \otimes \mathrm{Q}W' \multimap \mathcal{H} \otimes \mathcal{H}' \\ G \otimes G' \;&\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |w\rangle \otimes |w'\rangle \\ \quad \mathtt{do}\;\; G|w\rangle \otimes G'|w'\rangle\,, \end{array}\right. \end{aligned}$$

and the “pipe”-notation “ > “ for the sequential composition of maps may be declared as follows:

$$\begin{array}{l} \Phi_{12} : \mathrm{Q}W \multimap \mathcal{H}_2 \\ \Phi_{23} : \mathcal{H}_2 \multimap \mathcal{H}_3 \end{array} \quad\vdash\quad \begin{aligned} \Phi_{12} \;>\; \Phi_{23} \;&:\; \mathrm{Q}W_1 \multimap \mathcal{H}_3 \\ \Phi_{12} \;>\; \Phi_{23} \;&\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |w\rangle \\ \quad \mathtt{do}\;\; \Phi_{23}(\Phi_{12}|w\rangle)\,. \end{array}\right. \end{aligned}$$

In summary, to obtain neat pseudo-code we adopt and adapt traditional do-notation as follows:

Monadic declaration of a linear map $G : \mathrm{Q}W \multimap \mathcal{H}$ via the $\triangleright$-monad relativized to Q		
Traditional do-notation as in (127)	**for...do-notation** as in (132)	**for...do-notation** as used here
	$\left[\begin{array}{l} \mathtt{for}\;\; (w,1) \\ \quad \mathtt{do}\;\; G\lvert w\rangle \end{array}\right.$	$\left[\begin{array}{l} \mathtt{for}\;\; \lvert w\rangle \\ \quad \mathtt{do}\;\; G\lvert w\rangle \end{array}\right.$
$\lvert\psi\rangle \mapsto \left[\begin{array}{l} \mathtt{do} \\ \quad (w,1) \leftarrow \lvert\psi\rangle \\ \quad G\lvert w\rangle \end{array}\right.$	$\lvert\psi\rangle \mapsto \left[\begin{array}{l} \mathtt{for}\;\; (w,1)\;\; \mathtt{in}\;\; \lvert\psi\rangle \\ \quad \mathtt{do}\;\; G\lvert w\rangle \end{array}\right.$	$\lvert\psi\rangle \mapsto \left[\begin{array}{l} \mathtt{for}\;\; \lvert w\rangle\;\; \mathtt{in}\;\; \lvert\psi\rangle \\ \quad \mathtt{do}\;\; G\lvert w\rangle \end{array}\right.$

(251)

In linguistic generalization of this situation we therefore proceed to identify similarly suggestive verbalization of the structure maps of the other monadic effects from §2.3

`for`...`do`-notation

$$\mathrm{prog} \;:\; D \to \mathcal{E}D'$$

$$\mathtt{bind}^{\mathcal{E}}\mathrm{prog} \;:\; \mathcal{E}D \to \mathcal{E}D'$$

$$\mathtt{bind}^{\mathcal{E}}\mathrm{prog} \;\equiv\; \left[\begin{array}{ll} \mathtt{for} & \boxed{\mathtt{return}^{\mathcal{E}}_{D}(d)} \\ \mathtt{do} & \mathrm{prog}(d) \end{array}\right.$$

to be sugared
as per next table

$$\Phi \;:\; \mathcal{E}D, \qquad \mathrm{prog} \;:\; D \to \mathcal{E}D'$$

$$\phi \;>\; \mathtt{bind}^{\mathcal{E}}_{\mathrm{prog}} \;:\; \mathcal{E}D'$$

$$\phi \;>\; \mathtt{bind}^{\mathcal{E}}_{\mathrm{prog}} \;\equiv\; \left[\begin{array}{ll} \mathtt{for} & \boxed{\mathtt{return}^{\mathcal{E}}_{D}(b)} \quad \mathtt{in} \quad \Phi \\ \mathtt{do} & \mathrm{prog}(b) \end{array}\right.$$

This syntax is to closely reflect the fact that
– *for* an input of the form $\mathtt{return}^{\mathcal{E}}_{D}(d) : \mathcal{E}D$,
– which may appear as a *pure effect generator in* the input data
– the operation $\mathtt{bind}^{\mathcal{E}}_{\mathrm{prog}}$ *does* produce the output prog(d),
which prescription completely defines it, by linearity.

Sugared syntax for quantum measurement effects

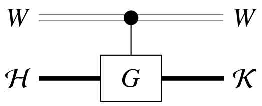
Quantization	$\lvert - \rangle$	⦂	$B \to \mathrm{Q}B$	
	$\lvert\, b\, \rangle$	$\equiv$	$\mathtt{return}^{\mathrm{Q}}_{B}(b)$	pure linearity
	$\lvert - \rangle \otimes \lvert - \rangle$	⦂	$B_1 \times B_2 \to \mathrm{Q}B_1 \otimes \mathrm{Q}B_2$	
	$\lvert b_1 \rangle \otimes \lvert b_2 \rangle$	$\equiv$	$\mathtt{return}^{\mathrm{Q(\text{-})} \otimes \mathrm{Q(\text{-})}}_{B_1 \times B_2}(b_1, b_2)$	pure entanglement
Quantum measurement	`definitely`	⦂	$\mathcal{H} \multimap \bigcirc_B \mathcal{H}$	
	`definitely` $\lvert\psi\rangle$	$\equiv$	$\mathtt{return}^{\bigcirc_B}_{\mathcal{H}}(\lvert\psi\rangle)$	pure indefiniteness
	`measure`	⦂	$\bigcirc_B \mathrm{Q}B \;\equiv\; \bigcirc_B \bigcirc_B \mathbb{1} \multimap \bigcirc_B \mathbb{1}$	
	`measure`	$\equiv$	$\mathrm{join}^{\bigcirc_B}_{\mathbb{1}}$	pure necessity
	`collapse`	⦂	$\mathrm{Q}B \multimap \bigcirc_B \mathbb{1}$	
	`collapse`	$\equiv$	`measure definitely`	returns collapsed state & lifts outcome into context
	`if measured` w `then` $\lvert\psi_w\rangle$	⦂	$\underset{W}{\bigcirc} \mathcal{H} \;\equiv\; (W \to \mathcal{H})$	
	`if measured` w `then` $\lvert\psi_w\rangle$	$\equiv$	$w \;\mapsto\; \lvert\psi_w\rangle$	condition quantum gate on measurement outcome

(252)

Coding quantum measurement. With (252) we obtain code expressing the quantum measurement typing from §2.4 (cf. p. 85) as shown on the right. Here (for W : $\mathrm{ClaType}^{\mathrm{fin}}$) $\texttt{collapse}_W$ (219) is the identity on underlying linear types, but understood as entering the measurement monad $\bigcirc_W$ and thereby lifting measurement results into the classical context, as witnessed by identifying:

$$\texttt{collapse} \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |w\rangle \\ \texttt{do} \left[\begin{array}{l}\texttt{if measured}\;\; w' \\ \texttt{then}\;\; \delta^{w'}_{w}\end{array}\right.\end{array}\right.$$

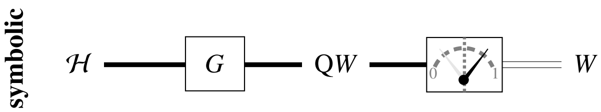
A quantum gate controlled (cf. p. 80) by a previous measurement result is thus coded as follows:



$$G_\bullet \;⦂\; \bigcirc_W \mathcal{H} \multimap \bigcirc_W \mathcal{K}$$

$$G_\bullet \;\equiv\; \left[\begin{array}{l} \texttt{for definitely}\,|\psi\rangle \\ \texttt{do} \left[\begin{array}{l}\texttt{if measured}\;\; w \\ \texttt{then}\;\; G_w|\psi\rangle\end{array}\right.\end{array}\right.$$

Quantum Gate followed by Measurement



$$\begin{tikzcd}[column sep=large]
\textbf{epistemic} & \mathrm{Q}V \underset{W}{\otimes} \mathbb{1} \arrow[r, "G \underset{W}{\otimes} \mathbb{1}"] \arrow[d, maps to, "\underset{W}{\bigoplus}"'] & \underset{W}{\square}\, \underset{W}{\mathbb{1}} \arrow[r, "\mathrm{obt}^{\square_W}_{\mathbb{1}_W}"] \arrow[d, maps to, "\underset{W}{\bigoplus}"'] & \underset{W}{\mathbb{1}} \arrow[d, maps to, "\underset{W}{\bigoplus}"'] \\
\textbf{effective} & \underset{W}{\bigcirc} \mathrm{Q}V \arrow[r, "\underset{W}{\bigcirc} G"] & \underset{W}{\bigcirc} \mathrm{Q}W \arrow[r, "\mathrm{join}^{\bigcirc_W}_{\mathbb{1}}", "\texttt{measure}"'] & \underset{W}{\bigcirc} \mathbb{1} \\
\textbf{computational} & \mathrm{Q}V \arrow[u, "\text{dynamic lifting}", "\mathrm{ret}^{\bigcirc_W}_{\mathrm{Q}V}"'] \arrow[r, "G" description, "\text{coherent quantum gate}"] & \mathrm{Q}W \arrow[u, "\mathrm{ret}^{\bigcirc_W}_{\mathrm{Q}W}", "\texttt{definitely}"'] \arrow[ur, "\texttt{collapse}"] &
\end{tikzcd}$$

$$\left[\begin{array}{l} \texttt{for}\;\; |v\rangle \\ \texttt{do}\;\; G|v\rangle \end{array}\right. \qquad \left[\begin{array}{l} \texttt{for}\;\; |w\rangle \\ \texttt{do} \left[\begin{array}{l}\texttt{if measured}\;\; w' \\ \texttt{then}\;\; \delta^{w'}_{w}\end{array}\right.\end{array}\right.$$

Remark 3.1 (Towards natural language)**.**

(i) The above sugared `for`...`do`-notation for classically-controlled quantum gates again neatly expresses the actual physical process in almost natural language: In general, the input state of a W-controlled quantum gate is itself a W-dependent quantum state $|\psi_w\rangle$, whence the epistemic declaration of $G_\bullet$ is of the form

$$w : W \quad \vdash \quad G_w \;\equiv\; \big(|\psi_w\rangle \;\mapsto\; G_w|\psi_w\rangle\big)\,,$$

but for describing the action of G_w on a generic state it does not matter whether this state carries a w-index, and this is what the `for`...`do`-notation reflects: It is sufficient to define G_w assuming that we are *definitely* presented with the state $|\psi\rangle$ (no matter the value of w), hence sufficient to define it *for* states of the form `definitely` $|\psi\rangle$.

(ii) With the components of the classically-controlled quantum gate themselves being coherent quantum gates, the latter may in turn be declared on basis states as before, which gives the following further nested declaration of a classically-controlled quantum gate, reducing to its component output states $(G_w|b\rangle \;:\; \mathcal{K})_{(w,b):W\times B}$:

Declaration of a measurement-controlled quantum gate in terms of its component values on each basis state $|b\rangle$ for each measurement result w.

$$G_\bullet \;⦂\; \bigcirc_W \mathrm{Q}B \multimap \bigcirc_W \mathcal{K}$$

$$G_\bullet \;\equiv\; \left[\begin{array}{l} \texttt{for definitely}\,|\psi\rangle \\ \texttt{do} \left[\begin{array}{l}\texttt{if measured}\;\; w \\ \texttt{then} \left[\begin{array}{l}\texttt{for}\;\; |b\rangle \;\; \texttt{in}\;\; |\psi\rangle \\ \texttt{do}\;\; G_w|b\rangle\end{array}\right.\end{array}\right.\end{array}\right.$$

(iii) The `return`-sugaring in the `for`...`do`-blocks is just that: The semantics of all notations in (251) are exactly identical. In particular, a declaration "`for` $|b\rangle$" has access to the actual variable b : B. For instance we can declare linear maps that duplicate the given basis states (needed in §3.2.3 below, for purposes of constructing "logical qbits") as follows

$$\mathrm{encode} \;⦂\; \mathrm{Q}W \multimap \mathrm{Q}(W \times W \times W)$$

$$\mathrm{encode} \;\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |w\rangle \\ \texttt{do}\;\; |w,\, w,\, w\rangle \end{array}\right.$$

3.2 Example Pseudocode

3.2.1 Standard QBit-gates

For reference we show a few basic quantum gates declared in `QS`-pseudocode, all of which examples of the general scheme (247), according to which a general linear map on QBit is coded by:

$$\begin{aligned} \Phi \;&\mathrel{\substack{\circ\\\circ}}\; \mathrm{QBit} \multimap \mathrm{QBit} \\ \Phi \;&\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\texttt{do}\;\; \Phi|b\rangle \end{array}\right. \end{aligned}$$

The quantum NOT-gate:

$$\begin{aligned} \mathrm{X} \;&\mathrel{\substack{\circ\\\circ}}\; \mathrm{QBit} \multimap \mathrm{QBit} \\ \mathrm{X} \;&\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\texttt{do}\;\; |1-b\rangle \end{array}\right. \end{aligned} \tag{253}$$

The CNOT-gate (17)

$$\begin{aligned} \mathrm{CNOT} \;&\mathrel{\substack{\circ\\\circ}}\; \mathrm{Q}(\mathrm{Bit}\times\mathrm{Bit}) \multimap \mathrm{Q}(\mathrm{Bit}\times\mathrm{Bit}) \\ \mathrm{CNOT} \;&\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b_1,\, b_2\rangle \\ \;\texttt{do}\;\; |b_1,\; b_1\,\texttt{xor}\,b_2\rangle \end{array}\right. \end{aligned} \tag{254}$$

The Hadamard gate:[35]

$$\begin{aligned} \mathrm{H} \;&\mathrel{\substack{\circ\\\circ}}\; \mathrm{QBit} \multimap \mathrm{QBit} \\ \mathrm{H} \;&\equiv\; \left[\begin{array}{l} \texttt{for}\;\; |b\rangle \\ \;\texttt{do}\;\; \tfrac{1}{\sqrt{2}}\big(|0\rangle + (-1)^b|1\rangle\big) \end{array}\right. \end{aligned} \tag{255}$$

The Bell state:

$$\begin{aligned} \mathrm{BellState} \;&\mathrel{\substack{\circ\\\circ}}\; \mathrm{QBit}\otimes\mathrm{QBit} \\ \mathrm{BellState} \;&\equiv\; \tfrac{1}{\sqrt{2}}(|0\rangle\otimes|0\rangle + |1\rangle\otimes|1\rangle) \end{aligned} \tag{256}$$

In typical discussion of QBit-circuits, the initial QBit-states are all assumed to be $|0\rangle$, and the Bell state (256) is prepared by sending $|0\rangle\otimes|0\rangle$ through the quantum circuit $(\mathrm{H}\otimes\mathrm{id}) > \mathrm{CNOT}$ (cf. the first step in the circuit shown on page 3). With the identification types available in `LHoTT` it is possible to construct a formal certificate that this indeed yields the intended state:

$$\texttt{verify\_Bell\_preparation} \quad : \quad \mathrm{BellState} \;=\; |0\rangle\otimes|0\rangle \;>\; (\mathrm{H}\otimes\mathrm{id}) \;>\; \mathrm{CNOT}$$

---

[35] The irrtraditional factor $1/\sqrt{2}$ in the Hadamard gate – whose implementation in a formal language like `LHoTT`, while certainly possible, opens a can of worms (cf. [TQP, pp. 71]) – has the purpose of making the map be unitary with respect to the canonical Hermitian inner product structure on QBit. But since we are not imposing the Hermitian structure in the QBit data type, for the time being, the factor could as well be omitted for ease of full formalization of the pseudo-code, at the small cost of picking up some irrelevant factors of 2 in subsequent expressions. For example, the quantum teleportation protocol §3.2.2 without these prefactors in H will not strictly reproduce the input state $|\psi\rangle$, but return it multiplied by 2 – which is physically still the same state, of course, up to normalization.

3.2.2 Quantum Teleportation Protocol

In combined exposition of QS-pseudocode and of the quantum teleportation protocol (as shown in the circuit diagram in page 3, originally due to [BE$^+$], see [NC00, §1.3.7][BEZ20, §3.3]) we narrate the logic of quantum teleportation by perpetually switching between natural and QS-language:

The punchline of quantum teleportation is to send a quantum state $|\psi\rangle$ (typically: a qbit) into a process "Alice"

$$|\psi\rangle \;>\; \mathrm{Alice}(\cdot)$$

which itself only records classical measurement results (concretely: a pair of bits):

$$\mathrm{Alice}(\cdot) \mathrel{\substack{\circ\\ \circ}} \overset{\substack{\textbf{quantum}\\ \textbf{input}}}{\mathrm{QBit}} \multimap \overset{\substack{\textbf{classical}\\ \textbf{output}}}{\underset{\mathrm{Bit}^2}{\bigcirc}\mathbb{1}}$$

and yet such that the transmission of this purely *classical* information $\bigcirc_{\mathrm{Bit}^2}$ to a further process "Bob":

$$\mathrm{Alice}(\cdot) \;>\; \mathrm{Bob}(\cdot)$$

allows the latter to re-construct a quantum state

$$\mathrm{Bob}(\cdot) \mathrel{\substack{\circ\\ \circ}} \underset{\mathrm{Bit}^2}{\bigcirc}\mathbb{1} \multimap \underset{\mathrm{Bit}^2}{\bigcirc}\mathrm{QBit}$$

which is *definitely* equal to the initial state (ie. independently of Alice's intermediate measurement results):

$$\mathrm{verify} \quad : \quad |\psi\rangle \;>\; \mathrm{Alice}(\cdot) \;>\; \mathrm{Bib}(\cdot) \quad \overset{?}{=} \quad \mathtt{definitely}_{\mathrm{Bit}^2}\, |\psi\rangle\,.$$

For this to really work we need to fill in one missing ingredient indicated by "$(\cdot)$", namely the two processes need to "share an entanglement source" up front, in that they need to share the two "halfs" of a Bell state pair of maximally entangle qbits (256), like this:

$$\mathrm{verify} \quad : \quad \left[\begin{array}{l} \mathtt{for}\;\; |\mathrm{bell}_A\rangle \otimes |\mathrm{bell}_B\rangle \;\;\mathtt{in}\;\; \mathrm{BellState} \\ \;\mathtt{do}\;\; |\psi\rangle \;>\; \mathrm{Alice}(\mathrm{bell}_A) \;>\; \mathrm{Bob}(\mathrm{bell}_B) \end{array}\right. \quad = \quad \mathtt{definitely}_{\mathrm{Bit}^2}\;\; |\psi\rangle\,.$$

Thus, the global structure of the quantum teleportation protocol is given by the following code:

$$\begin{array}{l} \mathrm{teleport} \mathrel{\substack{\circ\\ \circ}} \mathrm{QBit} \multimap \underset{\mathrm{Bit}^2}{\bigcirc}\mathrm{QBit} \\ \mathrm{teleport} \;\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |b\rangle \\ \;\mathtt{do}\;\; \left[\begin{array}{l} \mathtt{for}\;\; |\mathrm{bell}_1\rangle \otimes |\mathrm{bell}_2\rangle \;\;\mathtt{in}\;\; \mathrm{BellState} \\ \;\mathtt{do}\;\; |b\rangle \;>\; \mathrm{Alice}\big(|\mathrm{bell}_1\rangle\big) \;>\; \mathrm{Bob}\big(|\mathrm{bell}_2\rangle\big) \end{array}\right. \end{array}\right. \end{array} \tag{257}$$

and it remains to declare the sub-processes Alice and Bob.

The procedure of Alice's protocol is to

(1.) entangle the input state with the Bell state
(2.) feed the result through a suitable quantum gate and then
(3.) measure in the Bit^2-basis and return the measurement result

like this:

$$\begin{array}{l} \mathrm{Alice} \mathrel{\substack{\circ\\ \circ}} \mathrm{QBit} \multimap (\mathrm{QBit} \multimap \underset{\mathrm{Bit}^2}{\bigcirc}\mathbb{1}) \\ \mathrm{Alice} \;\equiv\; \left[\begin{array}{l} \mathtt{for}\;\; |\mathrm{bell}_1\rangle \\ \;\mathtt{do}\;\; \left[\begin{array}{l} \mathtt{for}\;\; |b\rangle \\ \;\mathtt{do}\;\; (|b\rangle \otimes |\mathrm{bell}_1\rangle) \;>\; \mathrm{CNOT} \;>\; (\mathrm{H} \otimes \mathrm{id}) \;>\; \mathtt{collapse} \end{array}\right. \end{array}\right. \end{array} \tag{258}$$

The crux is that with the classical information received from Alice, Bob can apply quantum gates to his part of the Bell-state *conditioned on* this classical information, like this:

$$\begin{aligned} \mathrm{Bob} &\;:\; \mathrm{QBit} \multimap \big(\underset{\mathrm{Bit}^2}{\bigcirc}\mathbb{1} \multimap \underset{\mathrm{Bit}^2}{\bigcirc}\mathrm{QBit}\big) \\ \mathrm{Bob} &\;\equiv\; \left[\begin{array}{ll} \texttt{for} & |\mathrm{bell}_2\rangle \\ \texttt{do} & \left[\begin{array}{l} \texttt{if measured}\;\; (b_1, b_2) \\ \texttt{then}\;\; |\mathrm{bell}_2\rangle \;>\; \mathrm{X}^{b_1} \;>\; \mathrm{Z}^{b_2} \end{array}\right. \end{array}\right. \end{aligned} \tag{259}$$

The categorical semantics of this code, when in turn expressed in string diagram notation, gives the usual circuit-diagram for the quantum teleportation protocol as shown on page 3. But now the correct encoding of the protocol becomes formally *verifiable*:

If these procedures Alice and Bob are correctly coded, then the quantum state which Bob re-constructs from his Bell-state is definitely equal to the one that Alice originally received (independent of the random measurement results that Alice obtained), and we will be able to certify this property at compile-time by constructing a term of the following identification-type:

$$\mathrm{verify} \quad : \quad \prod_{|\psi\rangle \,:\, \mathrm{QBit}} \big(\mathrm{teleport}\,|\psi\rangle \;=\; \texttt{definitely}\,|\psi\rangle\big) \tag{260}$$

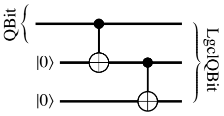
3.2.3 Quantum Bit Flip Code

Bit flip error correction as `QS`-pseudocode, is another simple but instructive example (cf. [NC00, §10.1.1]):

$$\mathrm{LgclQBit} \;:\; \mathrm{QuType} \qquad \mathrm{Syndrome} \;:\; \mathrm{ClaType}^{\mathrm{fin}}$$

$$\mathrm{LgclQBit} \;\equiv\; \mathrm{QBit}\otimes\mathrm{QBit}\otimes\mathrm{QBit} \qquad \mathrm{Syndrome} \;\equiv\; \mathrm{Bit}\times\mathrm{Bit}$$

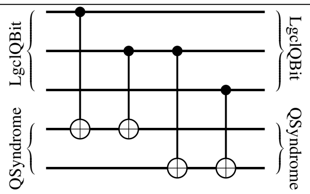
```
encode ⦂ $\mathrm{QBit} \multimap \mathrm{LgclQBit}$
encode ≡
    for $|b\rangle$
    do  $|b,b,b\rangle$
```



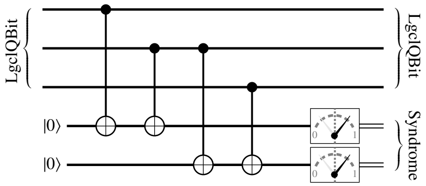
verify_circuit_encoding : encode $=$ $(-)\otimes|0,0\rangle \;>\; \mathrm{CNOT}\otimes\mathrm{id} \;>\; \mathrm{id}\otimes\mathrm{CNOT}$

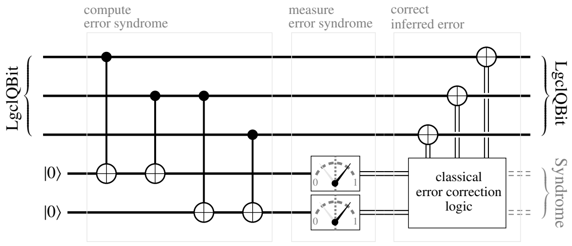
```
BitFlip ⦂ $\mathrm{Syndrome} \longrightarrow (\mathrm{LgclQBit} \multimap \mathrm{LgclQBit})$
BitFlip ≡
    if $(0,0)$ then $\mathrm{id}\otimes\mathrm{id}\otimes\mathrm{id}$
    if $(1,0)$ then $\mathrm{X}\otimes\mathrm{id}\otimes\mathrm{id}$
    if $(1,1)$ then $\mathrm{id}\otimes\mathrm{X}\otimes\mathrm{id}$
    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
        > collapse$_{\mathrm{Syndrome}}$
```





```
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 measure_syndrome$(|b_1,b_2,b_3\rangle)$
        do
            if measured $(s_1,s_2)$
            then $\mathrm{BitFlip}_{(s_1,s_2)}|\psi\rangle$
```

verify_error_correction : $(s_1,s_2 : \mathrm{Syndrome}) \rightarrow \big(\mathrm{encode} \;>\; \mathrm{BitFlip}_{s_1,s_2} \;>\; \mathrm{correct\_error} \;=\;$ `definitely` $\mathrm{encode}\big)$

Remark 3.2. 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.

Declarations.

Competing interests. The authors declare that they have no conflict of interest.
Data availability. There is no data associated with this work.

References

[Ab96] L. Abrams, *Two-dimensional topological quantum field theories and Frobenius algebra*, J. Knot. Theor. Ramifications **5** (1996), 569-587, [doi:10.1142/S0218216596000333].

[Aby09] 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].

[AC04] S. Abramsky and B. Coecke, *A categorical semantics of quantum protocols*, Proceedings of the 19th IEEE conference on Logic in Computer Science (LiCS'04). IEEE Computer Science Press (2004), [doi:10.1109/LICS.2004.1319636], [arXiv:quant-ph/0402130].

[AC07] S. Abramsky and B. Coecke, *Physics from Computer Science: a Position Statement*, Int. J. Unconventional Computing **3** 3 (2007), [ijuc-3-3-p-179-197].

[AC08] S. Abramsky and B. Coecke, *Categorical quantum mechanics*, in: *Handbook of Quantum Logic and Quantum Structures*, Elsevier (2008), [doi:10.1109/LICS.2004.1319636], [arXiv:0808.1023].

[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]

[AC95] M. Adelman and J. Corbett, *A Sheaf Model for Intuitionistic Quantum Mechanics*, Appl. Cat. Struct. **3** (1995), 79-104, [doi:10.1007/BF00872949].

[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].

[AG05] T. Altenkirch and J. Grattage, *A functional quantum programming language*, 20th Annual IEEE Symposium Logic in Computer Science (2005), 249-258, [doi:10.1109/LICS.2005.1], [arXiv:quant-ph/0409065].

[AG10] T. Altenkirch and A. Green, *The quantum IO monad*, Ch. 5 of: *Semantic Techniques in Quantum Computation* (2010), 173-205, [doi:10.1017/CBO9781139193313.006].

[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** (2014) 869893, [doi:10.1112/jtopol/jtt035], [arXiv:1403.4325].

[AMcD23] N. Arkor and D. McDermott, *The formal theory of relative monads*, [arXiv:2302.14014].

[ACh19] G. Arone and M. Ching, *Goodwillie Calculus*, in: *Handbook of Homotopy Theory*, Taylor & Francis (2019), [doi:10.1201/9781351251624], [arXiv:1902.00803].

[AA22] R. W. Arroyo and J. R. B. Arenhart, *Whence deep realism for Everettian quantum mechanics?*, Found. Phys. **52** (2022) 121, [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** (2015) 123010, [doi:10.1088/1367-2630/17/12/123010], [arXiv:1502.03450].

[At66] M. Atiyah, *K-theory and reality*, Quarterly J. Math. **17** (1966), 367-386, [doi:10.1093/qmath/17.1.367].

[Atk18] R. Atkey, *Syntax and semantics of quantitative type theory*, in: *33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS (2018), 56–65, [doi:10.1145/3209108.3209189].

[Att] S. Attal, *Lectures on Quantum Noises*, book project, [math.univ-lyon1.fr/~attal/chapters.html].

[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].

[Ba20] J. Baez, *Jordan algebras*, §4 of: *Getting to the Bottom of Noether's Theorem*, [arXiv:2006.14741].

[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`].

[BW85] M. Barr and C. Wells, *Toposes, Triples, and Theories*, Grundlehren der math. Wissenschaften **278**, Springer Berlin (1985); Reprints Theor. Appl. Categories **12** (2005), 1-287, [`tac:tr12`].

[Bas84] J. R. Bastida, *Field Extensions and Galois Theory*, Cambridge University Press (1984), [`doi:10.1017/CBO9781107340749`].

[Be69] J. Beck, *Distributive Laws*, in: Eckmann, B. (eds) Seminar on Triples and Categorical Homology Theory, Lecture Notes in Math. **80**, Springer, Berlin, Heidelberg (1969), [`doi:10.1007/BFb0083084`].

[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 Math. **47**, Springer, Berlin (1967), 1-77, [`doi:10.1007/BFb0074299`].

[Bé85] J. Bénabou, *Fibered Categories and the Foundations of Naive Category Theory*, J. 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`].

[BZ06] I. Bengtsson and K. Zyczkowski, *Geometry of Quantum States – An Introduction to Quantum Entanglement*, Cambridge University Press (2006), [`doi:10.1017/CBO9780511535048`].

[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`].

[BE$^+$] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres and W. K. Wootters, *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels*, Phys. Rev. Lett. **70** (1993) 1895, [`doi:10.1103/PhysRevLett.70.1895`].

[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), [`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, Berlin (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. M. Biedermann, private conversation with A. Joyal (2007).

[Bi94] G. M. Bierman, *On Intuitionistic Linear Logic*, DPhil dissertation, Cambridge University (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 and 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, and F. Wolter (eds.), *The Handbook of Modal Logic*, Studies in Logic and Practical Reasoning **3** (2007), [`cgi.csc.liv.ac.uk/ frank/MLHandbook`].

[Boa95] J. M. Boardman, *Stable Operations in Generalized Cohomology* in *Handbook of Algebraic Topology*, Oxford Univ. Press (1995), [`doi:10.1016/B978-0-444-81779-2.X5000-7`], [`ncatlab.org/nlab/files/Boardman-StableOperations.pdf`]

[Borg08] 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`].

[Born26] M. Born, *Quantenmechanik der Stoßvorgänge*, Zeitschrift Phys. **38** (1926), 803–827, [`doi:10.1007/BF01397184`].

[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`].

[Bra14] M. Brandenburg, *Tensor categorical foundations of algebraic geometry*, [`arXiv:1410.1716`].

[BBDRV09] H. J. Briegel, D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, *Measurement-based quantum computation*, Nature Phys. **5** (2009), 19-26, [`doi:10.1142/S0219749904000055`], [`arXiv:0910.1116`].

[BVS93] S. Brookes and K. Van Stone, *Monads and Comonads in Intensional Semantics*, technical report CMU-CS-93-10 (1993), [`cs.cmu.edu/~brookes/papers/MonadsComonads.pdf`].

[Bru14] 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`]

[BGL95] P. Busch, M. Grabowski, and P. J. Lahti, *Operational Quantum Physics*, Lecture Notes in Physics Monographs **31**, Springer, Berlin (1995), [`doi:10.1007/978-3-540-49239-9`].

[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*, Int. J. Theor. 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, and 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`].

[Cho22] S. Choe, *Quantum computing overview: discrete vs. continuous variable models*, [`arXiv:2206.07246`].

[Ch75] M.-D. Choi, *Completely positive linear maps on complex matrices*, Linear Algebra Appl. **10** 3 (1975), 285-290, [`doi:10.1016/0024-3795(75)90075-0`].

[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`].

[Co10] B. Coecke, *Quantum Picturalism*, Contemporary Physics **51** 1 (2010), 59-83, [`arXiv:0908.1787`], [`doi:10.1080/00107510903257624`].

[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`].

[CPaq08] B. Coecke and E. O. Paquette, *POVMs and Naimark's theorem without sums*, Electronic Notes Theor. Comp. Sci. **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*, Semantic Techniques in Quantum Computation, Cambridge University Press (2009), 29-69, [`doi:10.1017/CBO9781139193313.003`], [`arXiv:0904.1997`].

[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 and 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].

[Con11] 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]

[Cor20] 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*, Contemp. Math. **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 University Press (1973, 2016), [ISBN:9780691645926], [jstor:j.ctt13x0wwk].

[Di30] P. A. M. Dirac, *The Principles of Quantum Mechanics*, Oxford University Press (1930, 1981),
[ISBN:9780198520115].

[Di39] P. A. M. Dirac, *A new notation for quantum mechanics*, Math. Proc. Cambridge Philosophical Soc. **35** (1939), 416-418, [doi:10.1017/S0305004100021162].

[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].

[DP84] A. Dold and D. Puppe, *Duality, Trace and Transfer*, Proc. Steklov Inst. Math. **154** (1984), 85–103,
[maths.ed.ac.uk/ v1ranick/papers/doldpup2.pdf]

[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, Berlin (1966), 421-562, [doi:10.1007/978-3-642-99902-4].

[EM45] S. Eilenberg and S. MacLane, *General Theory of Natural Equivalences*, Transactions of the American Mathematical Society **58** 2 (1945), 231-294, [doi:10.1090/S0002-9947-1945-0013131-6], [jstor:1990284].

[EPR35] A. Einstein, B. Podolsky, and N. Rosen, *Can the Quantum-Mechanical Description of Physical Reality be Considered Complete?* Phys. Rev. **47** 10 (1935), 777-780, [doi:10.1103/PhysRev.47.777].

[Esp95] D. A. Espinosa, *Semantic Lego*, PhD thesis, Columbia University (1995),
[ncatlab.org/nlab/files/Espinosa-SemanticLego.pdf]

[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].

[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].

[Fr69] A. Frei, *Some remarks on triples*, Math. Zeitschrift **109** (1969), 269–272, [doi:10.1007/BF01110118].

[Fr64] P. Freyd, *Abelian Categories – An Introduction to the theory of functors*, Harper and Row (1964), Reprints in Theory and Applications of Categories **3** (2003), 23 - 164, [tac:tr3].

[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].

[FKRS22a] 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].

[FKRS22b] 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].

[Ge35] G. Gentzen, *Untersuchungen über das logische Schließen I*, Math. Zeitschrift **39** 1 (1935), 176–210, [doi:10.1007/BF01201353].

[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*, Theor. 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, [doi:10.1017/S0960129508007172], [arXiv:cs/0511006].

[Gr10] A. Green, *Towards a formally verified functional quantum programming language* (2010), [eprints:11457].

[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, [arXiv:1304.5485], [doi:10.1007/978-3-642-38986-3_10].

[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].

[Gri02] R. B. Griffiths, *Consistent Quantum Theory*, Cambridge University Press (2002), [doi:10.1017/CBO9780511606052].

[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].

[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].

[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].

[HZ11] T. Heinosaari and M. Ziman, *The Mathematical Language of Quantum Theory*, Cambridge University Press (2011), [`doi:10.1017/CBO9781139031103`].

[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`].

[HV12] C. Heunen and J. Vicary, *Lectures on categorical quantum mechanics* (2012), [`cs.ox.ac.uk/files/4551/cqm-notes.pdf`]

[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`].

[HHO03] M. Horodecki, P. Horodecki, and J. Oppenheim, *Reversible transformations from pure to mixed states, and the unique measure of information*, Phys. Rev. A **67** (2003) 062104, [`doi:10.1103/PhysRevA.67.062104`].

[HSS00] M. Hovey, B. Shipley, and J. Smith, *Symmetric spectra*, J. Amer. Math. Soc. **13** (2000), 149-208, [`doi:10.1090/S0894-0347-99-00320-3`], [`arXiv:math/9801077`].

[Ho19] M. Hoyois, *Topoi of parametrized objects*, Theory App. Categ. **34** (2019), 243-248, [`arXiv:1611.02267`], [`tac:34-09`].

[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*, J. Pure Appl. Algebra **105** (1995), 277-291, [`doi:10.1016/0022-4049(94)00153-7`].

[Hud96] P. Hudak, *Building Domain-Specific Embedded Languages*, ACM Computing Surveys **28** 4e (1996) 196–es, [`doi:10.1145/242224.242477`].

[Hud98a] P. Hudak, *Domain Specific Languages*, Chapter 3 in: *Handbook of Programming Languages*, Vol. III: *Little Languages and Tools*, MacMillan (1998), 39–60, [`cs448h.stanford.edu/DSEL-Little.pdf`]

[Hud98b] P. Hudak, *Modular Domain Specific Languages and Tools*, in: *Proceedings of Fifth International Conference on Software Reuse*, IEEE Computer Society Press (1998), [`doi:10.5555/551789.853532`].

[HHPW07] P. Hudak, J. Hughes, S. Peyton Jones, and P. Wadler, *A history of Haskell: being lazy with class*, History of Programming Languages (2007), 1-55, [`doi:10.1145/1238844.1238856`].

[HP07] M. Hyland and J. Power, *The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads*, Electronic Notes 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`].

[Ish95] C. Isham, *Lectures on Quantum Theory – Mathematical and Structural Foundations*, World Scientific, Singapore (1995), [`doi:10.1142/p001`].

[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*, J. 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`].

[JZ85] E. Joos and H. D. Zeh, *The emergence of classical properties through interaction with the environment*, Z. Physik B Condensed Matter **59** (1985), 223–243, [`doi:10.1007/BF01725541`].

[Jor27] P. Jordan, *Über eine neue Begründung der Quantenmechanik*, Zeitschrift Phys. **40** (1927), 809–838, [`doi:10.1007/BF01390903`].

[Jor32] P. Jordan, *Über eine Klasse nichtassociativer hyperkomplexer Algebren*, Nachr. Ges. Wiss. Göttingen (1932), 569-575, [`eudml:59403`].

[JvNW34] P. Jordan, J. von Neumann, and E. Wigner, *On an algebraic generalization of the quantum mechanical formalism*, Ann. Math. **35** (1934), 29-64, [`jstor:1968117`], [`doi:10.1007/978-3-662-02781-3_21`].

[Jo08] A. Joyal, *Notes on Logoi* (2008), [`ncatlab.org/nlab/files/JoyalOnLogoi2008.pdf`]

[KR97] R. V. Kadison and J. R. Ringrose, *Fundamentals of the theory of operator algebras*, Vol I *Elementary Theory*, Graduate Studies in Mathematics **15**, Amer. Math. Soc. (1997), [`ams:gsm-15`].

[Ka58] D. Kan, *Adjoint functors*, Trans. Amer. Math. Soc. **87** 2 (1958), 294-329, [`jstor:1993102`].

[Kar18] M. Karvonen, *The Way of the Dagger*, PhD thesis, Edinburgh (2018), [`arXiv:1904.10805`].

[Ke82] M. Kelly, *Basic concepts of enriched category theory*, London Math. Soc. Lec. Note Series **64**, Cambridge Univ. Press (1982), [`ISBN:9780521287029`]; Reprints in Theory Appl. Categories **10** (2005), 1-136, [`tac:tr10`].

[KNM10] V. M. Kendon, K. Nemoto, and W. J. Munro, *Quantum Analogue Computing*, Phil. Trans. R. Soc. A **368** 1924 (2010), 3621-3632, [`doi:10.1098/rsta.2010.0017`], [`arXiv:1001.2215`].

[KW20] S. Khatri and M. M. Wilde, *Principles of Quantum Communication Theory: A Modern Approach*, book draft, [`arXiv:2011.04672`].

[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`].

[Kl65] H. Kleisli, *Every standard construction is induced by a pair of adjoint functors*, Proc. Amer. Math. Soc. **16** AMS (1965), 544-546, [`jstor:2034693`].

[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`].

[KL98] E. Knill and R. Laflamme, *On the Power of One Bit of Quantum Information*, Phys. Rev. Lett. **81** (1998), 5672-5675, [`doi:10.1103/PhysRevLett.81.5672`], [`arXiv:quant-ph/9802037`].

[Kob97] S. Kobayashi, *Monad as modality*, Theoretical Computer Science **175** 1 (1997), 29-74, [`doi:10.1016/S0304-3975(96)00169-7`].

[KN63] S. Kobayashi and K. Nomizu, *Foundations of differential geometry*, Vol. 1 (1963), Vol. 2 (1969), Interscience Publishers; reprinted by Wiley Classics Library (1996), [`ISBN:978-0-470-55558-3`].

[Ko70] A. Kock, *Monads on symmetric monoidal closed categories*, Arch. Math. **21** (1970), 1-10, [`doi:10.1007/BF01220868`].

[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`].

[Ko20] A. Kornell, *Quantum Sets*, J. Math. Phys. **61** 102202 (2020) [`doi:10.1063/1.5054128`].

[KLM21] A. Kornell, B. Lindenhovius and M. Mislove, *Quantum CPOs*, EPTCS **340** (2021) 174-187 [`arXiv:2109.02196`] [`doi:10.4204/EPTCS.340.9`].

[Kr63] S. A. Kripke, *Semantical Analysis of Modal Logic I. Normal Modal Propositional Calculi*, Mathematical Logic Quarterly **9** 5-6 (1963), 67-96, [`doi:10.1002/malq.19630090502`].

[KPB15] N. Krishnaswami, P. Pradic, and N. Benton, *Integrating Dependent and Linear Types*, ACM SIGPLAN Notices **50** 1 (2015), 17–30, [`doi:10.1145/2775051.2676969`].

[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`].

[La09] K. Landsman, *The Born rule and its interpretation*, in: *Compendium of Quantum Physics*, Springer (2009), 64-70, [`doi:10.1007/978-3-540-70626-7_20`].

[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`].

[Lau93] J. Launchbury, *Lazy imperative programming*, Proceedings of ACM Sigplan Workshop on State in Programming Languages, Copenhagen (1993), [`launchbury.files.wordpress.com/2019/01/lazy-imperative-programming.pdf`]

[Law69a] W. Lawvere, *Adjointness in Foundations*, Dialectica **23** (1969) 281-296; reprinted in: Reprints in Theor. Appl. 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 Math. **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, [`arXiv:2110.02691`], [`doi:10.4230/LIPIcs.FSTTCS.2021.51`].

[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`].

[LHJ95] S. Liang, P. Hudak, and M. Jones, *Monad transformers and modular interpreters*, POPL '95 (1995), 333–343, [`doi:10.1145/199448.199528`].

[Li75] G. Lindblad, *Completely positive maps and entropy inequalities*, Commun. Math. Phys. **40** (1975), 147–151, [`doi:10.1007/BF01609396`].

[Li69] F. Linton, *An outline of functorial semantics*, in: *Seminar on Triples and Categorical Homology Theory*, Lecture Notes in Math. **80**, Springer, Berlin (1969), 7-52, [`doi:10.1007/BFb0083080`].

[Liu$^+$23] C. Liu, et al., *Quantum Memory: A Missing Piece in Quantum Computing Units*, [`arXiv:2309.14432`].

[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`]

[Lu18] M. Lundfall, *A diagram model of linear dependent type theory*, [`arXiv:1806.09593`].

[Luo94] Z. Luo, *Computation and Reasoning – A Type Theory for Computer Science*, Clarendon Press, Oxford (1994), [`ISBN:9780198538356`].

[Lu17] J. Lurie, *Higher Algebra* (2017), [`math.ias.edu/ lurie/papers/HA.pdf`]

[ML71/97] S. MacLane, *Categories for the Working Mathematician*, Graduate Texts in Mathematics **5** Springer, Berlin (1971, 1997), [`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, Berlin (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`].

[Mar66] J.-M. Maranda, *On Fundamental Constructions and Adjoint Functors*, Canadian Math. Bull. **9** 5 (1966), 581-591, [`doi:10.4153/CMB-1966-072-9`].

[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`]

[Mar09] F. Marty, *Des Ouverts Zariski et des Morphismes Lisses en Géométrie Relative*, Ph.D. Toulouse (2009), [`theses:2009TOU30071`], [`ncatlab.org/nlab/files/Marty-DesOuverts.pdf`]

[MaSi06] P. May and J. Sigurdsson, *Parametrized Homotopy Theory*, Mathematical Surveys and Monographs **132** AMS (2006), [`ams:surv-132`], [`arXiv:math/0411656`].

[McB16] C. McBride, *I Got Plenty o' Nuttin'*, in: *A List of Successes That Can Change the World,* Lecture Notes in Computer Science **9600**, Springer, Berlin (2016), [`doi:10.1007/978-3-319-30936-1_12`].

[McDU22] D. McDermott and T. Uustalu, *What Makes a Strong Monad?*, EPTCS **360** (2022), 113-133, [`doi:10.4204/EPTCS.360.6`], [`arXiv:2207.00851`].

[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*, The Bell System Technical J. **34** (1995), 1045-1079, [`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`].

[MR01] J. Mehra and H. Rechenberg, *The Probability Interpretation and the Statistical Transformation Theory, the Physical Interpretation, and the Empirical and Mathematical Foundations of Quantum Mechanics 1926-1932*, Part 1 in: *The Historical Development of Quantum Theory. Volume 6: The Completion of Quantum Mechanics, 1926-1941*, Springer, Berlin (2001), [`ISBN:978-0-387-98971-6`].

[Mel09] 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-categorielle-de-la-logique-lineaire`].

[Mer89] 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`].

[Mog89a] E. Moggi, *An abstract View of Programming Languages*, LFCS report ECS-LFCS-90-113 (1989), [`ncatlab.org/nlab/files/Moggi-AbstractView.pdf`]

[Mog89] E. Moggi, *Computational lambda-calculus and monads*, 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`].

[MEO21] B. Moon, H. Eades and D. Orchard, *Graded Modal Dependent Type Theory*, Programming Languages and Systems ESOP 2021, Lecture Notes in Computer Science **12648**, Springer, New York (2021), 462-490, [`doi:10.1007/978-3-030-72019-3_17`].

[Mo19] V. Moretti, *Fundamental Mathematical Structures of Quantum Theory*, Springer, New York (2019), [`doi:10.1007/978-3-030-18346-2`].

[MHP19] A. Müller-Hermes and C. Perry, *All unital qubit channels are 4-noisy operations*, Lett. Math. Phys. **109** (2019), 1–9, [`doi:10.1007/s11005-018-1104-x`], [`arXiv:1802.01337`].

[Mur14] D. Murfet, *Logic and linear algebra: an introduction*, [`arXiv:1407.2650`].

[MKZ13] M. Musz, M. Kuś, and K. Zyczkowski, *Unitary quantum gates, perfect entanglers, and unistochastic maps*, Phys. Rev. A **87** (2013) 022111, [`doi:10.1103/PhysRevA.87.022111`].

[TQP] D. J. Myers, H. Sati, and U. Schreiber, *Topological Quantum Gates in Homotopy Type Theory*, Communications in Mathematical Physics, **405** 172 (2024) [`arXiv:2303.02382`], [`doi:10.1007/s00220-024-05020-8`].

[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 Theor. Comp. Sci. **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*, Phys. Lett. A **308** (2003), 96–100, [`doi:10.1016/S0375-9601(02)01803-0`].

[NC00] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press (2000), [`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. Functional Programming **13** (2003), 747-796, [`doi:10.1017/S0956796802004495`].

[OP99] P. O'Hearn, D. J. Pym, *The Logic of Bunched Implications*, Bull. Symbolic Logic **5** 2 (1999), 215-244, [`doi:10.2307/421090`].

[OM16] J. N. Oliveira and V. C. Miraldo, *A practical approach to state-based system calculi*, J. Logical and Algebraic Methods in Programming **85** 4 (2016), 449-474, [`doi:10.1016/j.jlamp.2015.11.007`].

[Om94] R. Omnés, *The Interpretation of Quantum Mechanics*, Princeton University Press (1994), [`ISBN:9780691036694`].

[Par76] B. Pareigis, *Non-additive ring and module theory IV: The Brauer group of a symmetric monoidal category*, In: Zelinsky, D. (eds) Brauer Groups. Lecture Notes in Math. **549**, Springer, Berlin, Heidelberg (1976), 112–133, [`doi:10.1007/BFb0077339`].

[PPR19] D. Park, F. Petruccione, and J.-K. Rhee, *Circuit-Based Quantum Random Access Memory for Classical Data*, Scientific Reports **9** (2019) 3949, [`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`].

[Pa03] D. Pattinson, *An Introduction to the Theory of Coalgebras* (2003), [`nasslli.sitehost.iu.edu/2003/datas/DirkPattinson.pdf`]

[PRZ17] J. Paykin, R. Rand, and S. Zdancewic, *QWIRE: a core language for quantum circuits*, Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (2017), 846–858, [`doi:10.1145/3009837.3009894`].

[PZ19] J. Paykin and S. Zdancewic, *A HoTT Quantum Equational Theory*, QPL2019 (2019), [`arXiv:1904.04371`].

[PK23] M. Perone and G. Karachalias, *Composable Representable Executable Machines*, [`arXiv:2307.09090`].

[POM13] T. Petricek, D. Orchard, and A. Mycroft, *Coeffects: Unified Static Analysis of Context-Dependence*, Automata, Languages, and Programming. ICALP 2013, Lecture Notes in Computer Science **7966** Springer, New York (2013), [`doi:10.1007/978-3-642-39212-2_35`].

[PW93] S. L. Peyton Jones and P. Wadler, *Imperative functional programming*, POPL '93, Principles of programming languages (1993), 71-84, [`doi:10.1145/158511.158524`].

[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`].

[PP13] G. D. Plotkin and M. Pretnar, *Handling Algebraic Effects*, Logical Methods in Computer Science **9** 4 (2013) lmcs:705, [`doi:10.2168/LMCS-9(4:23)2013`], [`arXiv:1312.1399`].

[PS12] K. Ponto and M. Shulman, *Duality and traces in indexed monoidal categories*, Theor. Appl. Categ. **26** (2012) 23, [`tac:26-23`], [`arXiv:1211.1555`].

[PLMP03] D. Poulin, R. Laflamme, G. J. Milburn, and J. P. Paz, *Testing integrability with a single bit of quantum information*, Phys. Rev. A **68** (2003) 22302, [`doi:10.1103/PhysRevA.68.022302`], [`arXiv:quant-ph/0303042`].

[PW02] J. Power and H. Watanabe, *Combining a monad and a comonad*, Theoretical Computer Science **280** 1–2 (2002), 137-162, [`doi:10.1016/S0304-3975(01)00024-X`].

[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`].

[1] V. Pratt, *The second calculus of binary relations*, in *Mathematical Foundations of Computer Science 1993. MFCS 1993*, Lecture Notes in Computer Science **711**, Springer (1993) [`doi:10.1007/3-540-57182-5_9`]

[Pre04] J. Preskill, *Quantum Computation*, lecture notes (2004-), [`theory.caltech.edu/ preskill/ph229`].

[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`].

[Pu70] D. Pumplün, *Eine Bemerkung über Monaden und adjungierte Funktoren*, Math. Annalen **185** (1970), 329-337, [`eudml:161964`].

[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`]

[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`].

[Ra94] A. Ranta, *Type-theoretical grammar*, Oxford University Press (1994), [`ISBN:ISBN:9780198538578`].

[Ra12] K. S. Ratkovic, *Morita theory in enriched context*, PhD Thesis, Université de Nice (2012), [`hal:tel-00785301`], [`arXiv:1302.2774`].

[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*, 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`].

[Ri22a] M. Riley, *Linear Homotopy Type Theory*, talk at: *HoTTEST Event for Junior Researchers 2022* (Jan 2022) [`ncatlab.org/nlab/files/Riley-LHoTT-talk.pdf`] [`youtu.be/o2oWhHabjdM`]; based on: *A Bunched Homotopy Type Theory for Synthetic Stable Homotopy Theory*, PhD Thesis, Wesleyan University (2022), [`ir:3269`], [`doi:10.14418/wes01.3.139`].

[Ri22b] M. Riley, *Dependent Type Theories à la Carte*, talk at CQTS (Sep 2022) [`ncatlab.org/nlab/files/CQTS-InitialResearcherMeeting-Riley-220913.pdf`].

[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`].

[Ri21] F. Rios, *On a Categorically Sound Quantum Programming Language for Circuit Description*, PhD thesis, Dalhousie University (2021), [`hdl:10222/80771`].

[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`].

[Rob87] A. Robinson, *The extraordinary derived category*, Math. Z. **196** (1987), 231-238, [`doi:10.1007/BF01163657`].

[Ro15] N. J. Ross, *Algebraic and Logical Methods in Quantum Computation*, PhD thesis, Dalhousie University (2015), [`arXiv:1510.02198`].

[SN94] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, Cambridge University Press (1994), [`doi:10.1017/9781108587280`].

[Schau01] P. Schauenburg, *Turning Monoidal Categories into Strict Ones*, New York J. Math. **7** (2001), 257-265, [`nyjm:2001/7-16`].

[Schl07] M. Schlosshauer, *Decoherence and the Quantum-To-Classical Transition*, The Frontiers Collection, Springer, New York (2007), [`doi:10.1007/978-3-540-35775-9`].

[Schl19] M. Schlosshauer, *Quantum Decoherence*, Phys. Rep. **831** (2019), 1-57, [`doi:10.1016/j.physrep.2019.10.001`], [`arXiv:1911.06282`].

[SPWJ19] T. Schrijvers, M. Piróg, N. Wu, and M. Jaskelioff, *Monad transformers and modular algebraic effects: what binds them together*, Haskell 2019: Proceedings of the 12th ACM SIGPLAN International Symposium on Haskell (2019), 98–113, [`doi:10.1145/3331545.3342595`].

[Se13] G. J. Seal, *Tensors, monads and actions*, Theory Appl. Categ. **28** (2013), 403-434, [`tac:28-15`], [`arXiv:1205.0101`].

[Sel07] P. Selinger, *Dagger compact closed categories and completely positive maps*, Electronic Notes Theor. Comp. Sci. **170** (2007), 139-163, [`doi:10.1016/j.entcs.2006.12.018`].

[Se16] P. Selinger, *Introduction to Quipper*, talk at QPL2016 (2016), [`youtu.be/59frzb__Eqo`].

[Sa10] D. Samet, *S5 knowledge without partitions*, Synthese **172** (2010), 145–155, [`doi:10.1007/s11229-009-9469-0`].

[SS22] H. Sati and U. Schreiber, *Topological Quantum Programming in* `TED-K`, PlanQC **2022** 33 (2022), [`arXiv:2209.08331`], [`ncatlab.org/schreiber/show/Topological+Quantum+Programming+in+TED-K`].

[SS23a] H. Sati and U. Schreiber, *Anyonic defect branes in TED-K-theory*, Rev. Math. Phys **35** (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-Orb] H. Sati and U. Schreiber, *Proper Orbifold Cohomology*, [`arXiv:2008.01101`].

[SS21-EqB] H. Sati and U. Schreiber, *Equivariant Principal ∞-Bundles*, Cambridge University Press (2025, in print) [`arXiv:2112.13654`].

[SS23-EoS] H. Sati and U. Schreiber, *Entanglement of Sections*, [`arXiv:2309.07245`].

[SS23-QR] H. Sati and U. Schreiber, *Quantum and Reality* [arxIV:2311.11035]

[SS24-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, Oxford (1973), [`archive:logicalanalysiso0000sche`].

[SSt04] U. Schöpp and I. Stark, *A Dependent Type Theory with Names and Binding*, in: *Computer Science Logic. CSL 2004*, Lecture Notes in Computer Science **3210** (2004), 235–249, [`doi:10.1007/978-3-540-30124-0_20`].

[Sch13] U. Schreiber, *Differential cohomology in an ∞-topos*, [`arXiv:1310.7930`].

[Sch14a] U. Schreiber, *Quantization via Linear Homotopy Types*, talk notes, [`arXiv:1402.7041`].

[Sch14b] U. Schreiber, *Differential generalized cohomology in Cohesive homotopy type theory*, talk at Institut Henri Poincaré (2014), [`ncatlab.org/schreiber/files/SchreiberParis2014.pdf`]

[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`].

[ScSh03] S. Schwede and B. Shipley, *Stable model categories are categories of modules*, Topology **42** (2003), 103-153, [`doi:10.1016/S0040-9383(02)00006-X`].

[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`].

[Se12] P. Selinger, *Autonomous categories in which* $A \simeq A^*$, talk at QPL 2012, [`ncatlab.org/nlab/files/SelingerSelfDual.pdf`]

[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`].

[SJ08] P. W. Shor and S. P. Jordan, *Estimating Jones polynomials is a complete problem for one clean qubit*, Quantum Information and Computation **8** (2008) 681, [`doi:10.5555/2017011.2017012`], [`arXiv:0707.2831`].

[Sh12] M. Shulman, *Minicourse on Homotopy Type Theory*, University of Swansea (2012), [`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* $C^{\mathrm{op}} \to \mathrm{Cat}$, [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].

[StSt23] L. Stehouwer and J. Steinebrunner, *Dagger categories via anti-involutions and positivity* [arXiv:2304.02928]

[Str72] R. Street, *The formal theory of monads*, J. Pure Appl. Algebra **2** 2 (1972), 149-168, [doi:10.1016/0022-4049(72)90019-9].

[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].

[Th96] S. Thompson, *Haskell: the Craft of Functional Programming*, Addison-Wesley (1996), [haskellcraft.com/craft3e/Home.html].

[td87] T. tom Dieck, *Transformation Groups*, de Gruyter, Berlin (1987), [doi:10.1515/9783110858372].

[Tr92] A. S. Troelstra, *Lectures on Linear Logic*, CSLI Lectures Notes **29** (1992), [ISBN:0937073776].

[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]

[UV08] T. Uustalu and V. Vene, *Comonadic Notions of Computation*, Electronic Notes Theor. Comp. Sci. **203** 5 (2008), 263-284, [doi:10.1016/j.entcs.2008.05.029].

[Va15] M. Vékár, *A Categorical Semantics for Linear Logical Frameworks*, in *Foundations of Software Science and Computation Structures, FoSSaCS 2015*, Lecture Notes in Computer Science **9034**, Springer, New York (2015), [doi:10.1007/978-3-662-46678-0_7], [arXiv:1501.05016].

[Va17] M. Vékár, *In Search of Effectful Dependent Types*, PhD Thesis, Oxford University (2017), [arXiv:1706.07997], [uuid:e91e19b3-7e10-4fda-9433-f23b469e4049].

[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].

[VZ14] B. Valiron and S. Zdancewic, *Finite Vector Spaces as Model of Simply-Typed Lambda-Calculi*, in: *Proc. of ICTAC'14*, Lecture Notes in Computer Science **8687**, Springer (2014), 442-459, [doi:10.1007/978-3-319-10882-7_26], [arXiv:1406.1310], [cs.bham.ac.uk/ drg/bll/steve.pdf]

[vWe] J. van de Wetering, *ZX-calculus for the working quantum computer scientist*, [arXiv:2012.13966].

[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].

[Vic11] J. Vicary, *Categorical Formulation of Finite-Dimensional Quantum Algebras*, Commun. Math. Phys. **304** (2011), 765–796, [doi:10.1007/s00220-010-1138-0].

[Vo10] V. Voevodsky, *Univalent Foundations Project*, grant proposal application (2010), [ncatlab.org/nlab/files/Voevodsky-UFP2010.pdf]

[vN30] J. von Neumann, *Allgemeine Eigenwerttheorie Hermitescher Funktionaloperatoren*, Math. Ann. **102** (1930), 49–131, [doi:10.1007/BF01782338].

[vN32] J. von Neumann, *Mathematische Grundlagen der Quantenmechanik* (1932)
[doi:10.1007/978-3-642-96048-2];
Mathematical Foundations of Quantum Mechanics, Princeton University Press (1955), [doi:10.1515/9781400889921].

[vW51] G. H. von Wright, *An Essay in Modal Logic*, North-Holland Publishing, Amsterdam (1951), [`philpapers:VONA EI-2`].

[Wa90] P. Wadler, *Comprehending Monads*, Conference on Lisp and functional programming, ACM Press (1990), [`doi:10.1145/91556.91592`].

[Wa93] P. Wadler, *A syntax for linear logic*, Ninth International Conference on the Mathematical Foundations of Programming Semantics, Lecture Notes in Computer Science **802** Springer, Berlin (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`].

[Wh12] N. Wheeler, *Generalized Quantum Measurement* (2012), [`ncatlab.org/nlab/files/Wheeler-GeneralizedQuantumMeasurement.pdf`]

[Wil13] M. M. Wilde, *Quantum Information Theory*, Cambridge University Press (2013), [`doi:10.1017/CBO9781139525343`], [`arXiv:1106.1445`].

[Wi20] A. Wilson, *The Nature of Contingency: Quantum Physics as Modal Realism*, Oxford University Press (2020), [`ISBN:9780198846215`].

[Wi22] S. Winitzki, *The Science of Functional Programming – A tutorial with examples in Scala* (2022), [`github:sofp`], [`leanpub:sofp`].

[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`].

[WZ82] W. Wooters and W. Zurek, *A single quantum cannot be cloned*, Nature **299** (1982), 802-803, [`doi:10.1038/299802a0`].

[XCGX23] J. Xuereb, S. Campbell, J. Goold, and A. Xuereb, *DQC1 as an Open Quantum System*, Phys. Rev. A **107** (2023) 042222, [`doi:10.1103/PhysRevA.107.042222`], [`arXiv:2209.03947`].

[Ya19] R. Yates, *Improving Haskell Transactional Memory*, PhD thesis, University of Rochester (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/`].

[Zeh70] H. D. Zeh, *On the interpretation of measurement in quantum theory*, Found Phys **1** (1970), 69–76, [`doi:10.1007/BF00708656`].

[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`].

[Zu81] W. Zurek, *Pointer basis of quantum apparatus: Into what mixture does the wave packet collapse?*, Phys. Rev. D **24** (1981), 1516-1525, [`doi:10.1103/PhysRevD.24.1516`].

[ZB04] K. Zyczkowski and I. Bengtsson, *On Duality between Quantum Maps and Quantum States*, Open Systems & Information Dynamics **11** 01 (2004), 3-42, [`doi:10.1023/B:OPSY.0000024753.05661.c2`].