Quantum Orchestras: a Concrete Semantics for Recursive Hybrid Programs

ALEX RICE, The University of Edinburgh, United Kingdom
DOMINIK LEICHTLE, The University of Edinburgh, United Kingdom
KIM WORRALL, The University of Edinburgh, United Kingdom
ROBERT I. BOOTH, University of Oxford, United Kingdom

Many production quantum programming languages represent hybrid quantum computations by extending a classical base language with a quantum effect, where qubits are addressed by reference, and quantum operations are understood to mutate some external quantum state. However, the semantics of this view of quantum computation remains underdeveloped, especially when the language allows mid-circuit measurements and non-termination.

In this work, we provide a general method for building denotational semantics for such languages, by defining the *quantum orchestra* monad, which precisely captures this style of quantum effect. The monad has a concrete presentation, being based on the formalism of quantum instruments, a common tool in quantum information theory for capturing the action of a quantum process along with its classical outcomes. It acts on the category **DCPO**, and so enables the interpretation of divergent hybrid programs.

The quantum orchestra monad serves as a natural extension of both the classical state monad and the probabilistic powerdomain monad. We investigate some of the subtleties present when trying to naïvely extend these definitions to the quantum non-commutative case.

1 Introduction

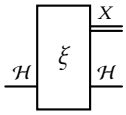
The field of quantum computing is quickly progressing from small scale experiments in physics labs to increasingly capable industrial quantum computers. As these systems grow in scale, the software used to control and program them is becoming correspondingly more complex. This motivates the development of mathematical models that are both expressive and conceptually clear, supporting the design and analysis of modern quantum programming languages.

Traditionally, quantum programs have been expressed using the circuit model [12]: a static list of quantum gates is applied sequentially to an input quantum state, which is then measured at the end of the computation. While the semantics of circuits are well understood and their static nature makes it possible for them to be heavily optimised, programming directly in terms of circuits provides little support for programs that adapt their quantum behaviour in response to intermediate measurement outcomes.

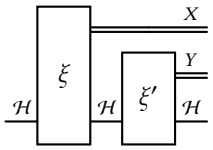
More recently, there has been increased interest in hybrid quantum programs, which combine quantum operations with classical computation and control flow (see for example IBM introducing classical registers in OpenQASM 3 [10], Quantinuum's HUGR [30], QIRO [22] which is used in the Pennylane toolkit [4], and QIR [37]). This hybrid functionality is essential for describing many aspects of quantum computing, including variational algorithms [8], measurement-based quantum computing [38], repeat-until-success circuits [1], error mitigation [48], qubit teleportation [13], iterative phase estimation [18], circuit knitting [36], and quantum error correction (e.g. [40]).

At a high level, many such languages follow the quantum random-access machine (QRAM) model of quantum computation [29]: a classical host program dynamically controls a quantum processor, issuing quantum commands and observing the outcomes of measurements. This view is particularly natural for languages that extend a classical base language with operations for

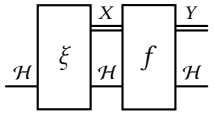
Authors' Contact Information: Alex Rice, The University of Edinburgh, United Kingdom, alex.rice@ed.ac.uk; Dominik Leichtle, The University of Edinburgh, United Kingdom, dominik.leichtle@ed.ac.uk; Kim Worrall, The University of Edinburgh, United Kingdom, kim.worrall@ed.ac.uk; Robert I. Booth, University of Oxford, United Kingdom, firstname.lastname@cs.ox.ac.uk.



(a) A quantum instrument $\xi \in Q(X)$.



(b) Naïve composition of $\xi \in Q(X)$ and $\xi' \in Q(Y)$ yielding $Q(X \times Y)$.



(c) Monadic composition of $\xi \in Q(X)$ and $f : X \to Q(Y)$ yielding $Q(Y)$.

Fig. 1. Sequential composition of quantum instruments. A fixed composition of $\xi \in Q(X)$ with $\xi' \in Q(Y)$ cannot express measurement-dependent control. The relevant operation is Kleisli composition, composing ξ with a continuation $f : X \to Q(Y)$.

manipulating quantum state, such as Guppy [31] and Silq [5]. In this setting, the classical program manipulates references to quantum resources whose underlying state evolves as the computation proceeds.

Despite its practical importance, the denotational semantics of this QRAM-style programming model remains comparatively underdeveloped. Unlike static circuits, hybrid quantum programs combine ordinary classical computation with effectful interaction with an evolving quantum state. Each interaction with the quantum processor may both update the quantum state and return classical information to the host program, which can then influence the rest of the computation. The challenge is to model this interaction compositionally while accommodating richer features of the classical host such as recursion.

A simple example is the reset operation on a single qubit, which maps every input state to $|0\rangle$. Operationally, reset can be implemented by measuring the qubit and then conditionally applying an X gate if the outcome was $|1\rangle$:

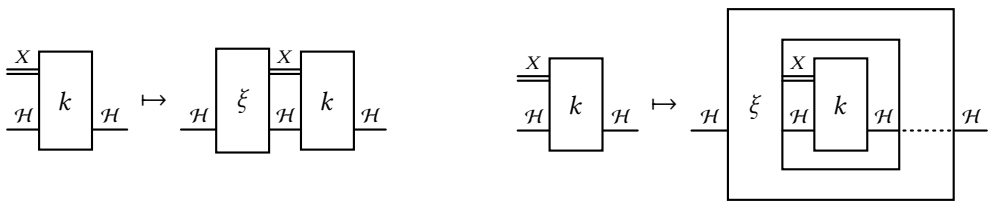
$$\textsc{reset} := \mathsf{let}\, b = \textsc{meas}(q)\, \mathsf{in}\, \{\ \mathsf{if}\, b\, \mathsf{then}\, \mathrm{X}(q)\ \}$$

This program exhibits measurement feedforward: a classical measurement outcome determines which quantum operation is performed next. Although the quantum state transformation induced by measurement can be described by a single channel acting on the quantum system, the program itself cannot be modelled by simply composing such channels, because the choice of the next channel depends on the classical outcome of the previous interaction.

The standard mathematical formalism for this kind of classical-quantum interaction is the *quantum instrument*. Intuitively, an instrument assigns to each possible classical outcome a corresponding quantum state transformation. It therefore records both parts of an interaction with a quantum processor: the classical information returned to the host program, and the effect of that interaction on the quantum state.

This also explains why we require a monadic composition for instruments—see Fig. 1. If $\xi \in Q(X)$ is an instrument with classical output X, then the next step of the program need not be a fixed instrument $\xi' \in Q(Y)$. Rather, it may depend on the observed outcome $x \in X$, and is therefore described by a function $f : X \to Q(Y)$. Sequential composition of hybrid quantum computations is thus Kleisli composition: first run ξ, observe its classical output, and then continue with the instrument selected by that output. The reset example above is the simplest instance of this pattern: the result of measurement determines whether the subsequent operation is the identity or the X gate.

We argue that quantum instruments are the fundamental primitive of QRAM-style quantum programming, and give rise to a natural computational effect: pure computations are classical, while



(a) Quantum orchestras capture precomposition with a quantum instrument.

(b) The orchestra can access all interfaces of k, but channels slide along the $\mathcal{H}$-wire on the right.

Fig. 2. Two graphical depictions of the action of a quantum orchestra ξ on a continuation k.

effectful computations interact with the quantum processor through instruments. In fact, we take a further step: we consider not only programs which condition on the results of measurement, but also those with general recursion, whose termination is dependent on the results of measurement. We capture this effect by generalising quantum instruments to our *quantum orchestra* monad over directed complete partial orders, obtaining domain-theoretic semantics of hybrid quantum–classical programs.
Concretely, the quantum orchestra monad supports:

- arbitrary unitary evolution of a quantum state;
- mid-circuit measurement and measurement-dependent control flow;
- general recursion, via a fixpoint combinator obtained by defining the monad over the category **DCPO** of directed-complete partial orders;
- qubit allocation, by presenting the monad as a parameterised strong monad [3] over the category $\mathbf{W}^*$ of von Neumann algebras and quantum channels.

To demonstrate the utility of our monad, we give denotational semantics for a prototype hybrid language. Rather than tailoring the semantics to an existing language, we develop the language throughout the paper, incrementally adding common hybrid features, and demonstrating in turn how each of these features is modelled by the monad. This presents the quantum orchestra monad as a flexible tool for giving semantics to a variety of hybrid languages, rather than a model for one specific language.

Remark. While quantum evolution is frequently thought of as a forward-progression of quantum states in the Schrödinger picture, it is oftentimes algebraically elegant to adopt the Heisenberg perspective in which quantum channels pull measurements backwards. The design of the quantum orchestra monad heavily draws inspiration from this paradigm and represents computations in *continuation passing style*, pulling continuations backwards through quantum instruments. For finite classical outcomes, quantum orchestras act on continuations, as expected, through precomposition with a quantum instrument (Fig. 2a). Clearly, the treatment of unbounded loops however requires the generalisation to infinite classical outcomes. In this case, the action of quantum orchestras becomes slightly more intricate (Fig. 2b), while retaining desirable features such as compositionality with subsequent computations.

Related work. Existing approaches to the semantics of hybrid quantum computation differ in how classical control and quantum effects are integrated. Measurement has been treated as an algebraic effect over combined quantum-classical programming language [45], added as a classical effect to a purely quantum programming language [20], and monadically without a denotational semantics [2]. Quantum processes have been given categorical models, such as Selinger's CPM construction [43],

which models mixed-state quantum evolution and measurement via completely positive maps in dagger compact categories, or Huot and Staton's result showing that the category of completely positive trace-preserving maps is a canonical completion of the category of isometries [21]. The quantum lambda calculus [44] uses linear logic to model hybrid quantum computing. Monads have been used to describe circuit-generation in Quipper [19], and extended to allow mid-circuit measurement [39, 42]. In contrast, we model quantum computation itself as a monadic effect over a purely classical language, using quantum instruments directly as the denotational structure. Closest to our work, Jia *et al.* [23] describe an extension of the probabilistic fixpoint calculus by quantum effects, using the theory of *Kegelspitzen* [28] to link the classical part of the language modelled using a valuations monad, and the quantum part modelled with von Neumann algebras.

Similarly to the monad proposed in this paper, there exist other recent works [6, 15] that construct monads based on quantum instruments. While our monad acts on **DCPO** and can thus be seen as a non-commutative quantum generalisation of the probabilistic powerdomain monad [24], their monads act on **Meas**, the category of measurable spaces, and are therefore closer to generalisations of the classical Giry monad [17]. The previously proposed monads could hence not directly serve to give semantics to recursive programs. Interestingly, however, all of the constructions, including ours, coincide in the special case of finite classical outcomes.

Outline of this paper. In Section 2, we recall the necessary background from the literature regarding operator algebra, domain theory, and the study of quantum instruments. Section 3 then introduces our toy language by adding quantum effects successively to a pure classical base language, while introducing quantum instruments as the natural tool to model its semantics. At the core of our contributions, Section 4 eventually presents and discusses the *Quantum Orchestra Monad* on **DCPO** as a way to model diverging quantum programs, with proofs of its well-definedness and of the monad laws given in Section 5. We conclude with Section 6 which clarifies connections to other known constructions, discusses future extensions to the current work, and discusses some open problems.

2 Background

Operator algebras, and in particular von Neumann algebras, are a well-established and standard tool in the mathematical description of quantum information and its physical evolution. While it is largely equivalent to the usual Hilbert space formulation, its power, especially in infinite-dimensional settings, will become apparent later on. The strong order-structure of von Neumann algebras, as witnessed by Theorem 10, will be one of the main pillars of our results. In the following, we give the most important definitions and known results about W*-algebras and their order-theoretic structure for our purposes.

2.1 W*-algebras

A *C*-algebra* is a Banach $*$-algebra $\mathscr{A}$ that satisfies the C*-identity:

$$\forall a \in \mathscr{A}. \quad \|aa^*\| = \|a\|\|a^*\|$$

A *W*-algebra* (also known as von Neumann algebra) is a C*-algebra $\mathscr{A}$ which admits a topological predual $\mathscr{A}_*$. The weak-* topology induced on $\mathscr{A}$ by the functionals in $\mathscr{A}_* \subseteq (\mathscr{A}_*)^{**} = \mathscr{A}^*$ is also called the ultraweak topology. A C*-algebra is called *unital* if it contains an identity; in this work, *all* C*-algebras (including W*-algebras) are assumed to be unital.

Example 1. Given any Hilbert space $\mathcal{H}$, the W*-algebra $\mathcal{B}(\mathcal{H})$ is the algebra of bounded linear operators on $\mathcal{H}$. It plays a particularly important role as the algebra generated by all measurement effects on $\mathcal{H}$. Many of the W*-algebras used in this work will be of this form.

The set of self-adjoint elements of $\mathscr{A}$, *i.e.*, all $a \in \mathscr{A}$ that satisfy $a = a^*$, forms a partially ordered vector space where $a \geq 0$ if and only if $a = b^*b$ for some $b \in \mathscr{A}$ (this order is referred to as the *Löwner order*). The set $\mathscr{A}^+ = \{a \in \mathscr{A} \mid a \geq 0\}$ of positive elements of $\mathscr{A}$ forms a convex cone.

Definition 2. Let $\mathscr{A}, \mathscr{B}$ be W*-algebras, $\Phi : \mathscr{B} \to \mathscr{A}$ be a linear map, and $\Phi^{(n)}$ its n-th matrix amplification. Then Φ is called

(1) *normal* (n) if it is ultraweakly continuous;
(2) *positive* (P) if $\Phi(b) \geq 0$ for all $b \in \mathscr{B}^+$;
(3) *completely positive* (CP) if $\Phi^{(n)}$ is positive for all $n \in \mathbb{N}$;
(4) *unital* (U) if $\Phi(1_{\mathscr{B}}) = 1_{\mathscr{A}}$;
(5) *subunital* (SU) if $\Phi(1_{\mathscr{B}}) \leq 1_{\mathscr{A}}$.

Proposition 3 ([9]). *The exists a symmetric monoidal category* $\mathbf{W^*_{nCPSU}}$ *where the objects are W*-algebras, the morphisms are normal completely positive subunital (nCPSU) maps, and the monoidal structure is given by the spatial von Neumann tensor product (see [41, Definition 1.22.10] for a full definition). In the following we will also denote this category simply as* $\mathbf{W^*}$.

Whereas the reader might associate quantum channels in the Schrödinger picture with completely positive trace-preserving (CPTP) or completely positive trace-non-increasing (CPTNI) maps, it will be convenient to switch to the (dual) Heisenberg picture in which trace-non-increasingness is replaced by subunitality. In this sense, one might think of the category $\mathbf{W^*}^{\mathrm{op}}$ as the category of quantum channels.

Example 4. Two particularly important channels are the identity and the zero channel. Given a W*-algebra $\mathscr{A}$, the *identity channel* on $\mathscr{A}$ is given by the identity map on $\mathscr{A}$, and denoted $\mathrm{id}_{\mathscr{A}}$. It is normal, completely positive, and unital. For W*-algebras $\mathscr{A}, \mathscr{B}$, the *zero channel* from $\mathscr{A}$ to $\mathscr{B}$ is given by the nCPSU map $0_{\mathscr{B}\mathscr{A}} : \mathscr{B} \to \mathscr{A},\ b \mapsto 0$.

Example 5 (Isometric quantum channels)**.** Let $V : \mathcal{H} \to \mathcal{K}$ be an isometry. Then the channel

$$\overline{V} : \mathcal{B}(\mathcal{K}) \to \mathcal{B}(\mathcal{H}), \quad M \mapsto V^\dagger M V,$$

is normal, completely positive, and unital. It is often convenient to obtain quantum channels in this way. Common examples include unitary channels (when V is a unitary), see Section 3.2, and qubit allocation, as used in Section 3.4.

For a detailed introduction to W*-algebras beyond these basics, we refer to [41].

2.2 Directed complete partial orders

We refer the reader to [9] for a more detailed discussion of the order structure of W*-algebras.

Definition 6 (Directed complete partial order)**.** Let X be a set and $\leq$ be a partial order on X.

(1) A subset $Y \subseteq X$ is called *directed* if for all $x, y \in Y$ there exists $z \in Y$ such that $x, y \leq z$.
(2) X is called *directed complete* if every directed subset has a supremum.
(3) X is called *pointed* if it has a least element $\bot$.

In the following, directed complete partially ordered sets will be abbreviated as *dcpo*.

Definition 7 (Scott-continuity)**.** Let X, Y be dcpos and $f : X \to Y$. Then f is called *Scott-continuous* if it preserves the suprema of all directed subsets. All Scott-continuous functions are *monotone*, *i.e.*, $f(x) \leq f(y)$ whenever $x \leq y$.

Definition 8 (Category of dcpos)**.** There is a category **DCPO** whose objects are dcpos and whose morphisms are Scott-continuous maps.

Theorem 9 (Kleene fixed-point theorem [46, 47])**.** *Let $f : X \to X$ be a Scott-continuous function on a pointed dcpo X. Then f has a least fixed point $\mathit{fix}\, f$ given by*

$$\mathit{fix}\, f = \sup_n f^n(\bot).$$

Moreover, the function $\mathit{fix} : (X \to X) \to X$ is itself continuous.

Theorem 9 is an essential and well-established tool to give meaning to diverging programs, and one of the reasons why domain theory became an important field of study for denotational semantics.

For W*-algebras $\mathscr{A}, \mathscr{B}$, we define a partial order $\leq$ on the maps in $\mathbf{W^*}(\mathscr{B}, \mathscr{A})$ by letting $\Phi \geq \Psi$ if and only if $\Phi - \Psi$ is completely positive.

Theorem 10 ([9])**.** *Let $\mathscr{A}, \mathscr{B}$ be W*-algebras. Then, $\mathbf{W^*}(\mathscr{B}, \mathscr{A})$ with the partial order $\leq$ is a pointed dcpo with the zero map as its least element. In fact, the category $\mathbf{W^*}$ is $\mathbf{DCPO}$-enriched.*

2.3 Quantum instruments

Measurements differ from ordinary quantum evolutions in that they produce both a quantum output and a classical outcome. In the Heisenberg picture, this means that, rather than a single normal completely positive subunital map describing the transformation of observables, one has a family of such maps indexed by the possible outcomes. Each map describes the transformation of observables conditional on a particular outcome, while their sum recovers the unconditional evolution obtained by disregarding the measurement result. This leads to the notion of a quantum instrument.

Definition 11 (Finite quantum instrument)**.** Let X be a set and $\mathscr{A}, \mathscr{B}$ be W*-algebras. Then, a (finite) quantum instrument from $\mathscr{A}$ to $\mathscr{B}$ is a collection of normal completely positive subunital maps $\{\Psi_x : \mathscr{B} \to \mathscr{A}\}_{x \in X}$ such that only finitely many maps Ψ_x are non-zero and such that $\sum_{x \in X} \Psi_x$ is subunital.

We interpret the set X as the set of possible measurement outcomes. In the concrete case where $\mathscr{A} = \mathcal{B}(\mathcal{H})$ and $\mathscr{B} = \mathcal{B}(\mathcal{H}')$, each map $\Psi_x : \mathcal{B}(\mathcal{H}') \to \mathcal{B}(\mathcal{H})$ pulls back observables on the output system $\mathcal{H}'$ to the input system $\mathcal{H}$. Given an observable $A \in \mathcal{B}(\mathcal{H}')$ and quantum state $\rho \in \mathcal{B}(\mathcal{H})_*$, $\rho(\Psi_x(A))$ describes the conditional contribution of outcome x to the unconditional expectation value $\sum_{x \in X} \rho(\Psi_x(A))$ of A when the measurement result is discarded. In particular, $\Psi_x(1)$ is the effect associated to outcome x, since $\rho(\Psi_x(1))$ is the probability of observing x.

Definition 12 (Dirac quantum instrument)**.** For any $x \in X$ and $\Psi \in \mathbf{W^*}(\mathscr{B}, \mathscr{A})$, we define the *Dirac quantum instrument* $\delta(x, \Psi)$ as

$$\delta(x, \Psi)_{x'} = \begin{cases} \Psi, & \text{if } x = x', \\ 0, & \text{otherwise.} \end{cases}$$

We record the following straightforward fact about finite quantum instruments.

Lemma 13. *All finite quantum instruments are finite sums of Dirac instruments.*

Going beyond the finite-outcome case, quantum instruments with continuous classical outcomes have been formally defined and extensively studied in a measure-theoretic context [6, 11, 15]. In the context of domain theory however, while a form of quantum instrument has been considered in the finite case [9], this work is the first to give a description in the case of infinite classical outcomes, to the best of the authors' knowledge.

2.4 Monads

Monads are a central tool for understanding *effectful* computation, functions which have side effects capturing behaviour other than returning a value. They have been used both as a semantic tool and internally to programming languages to embed effects in pure languages. We recall their definition.

Definition 14 (Monad)**.** A monad is a triple (T, η, μ), where T is an endofunctor on a category $\mathcal{C}$, the unit η is a natural transformation from the identity functor on $\mathcal{C}$ to T, and the multiplication μ is a natural transformation from T^2 to T subject to the usual laws.

Instead of giving a multiplication operation, monads can be defined in terms of *Kleisli extension*, which maps a morphism $f \in \mathcal{C}(X, T(Y))$ to a function $f^\sharp : \mathcal{C}(T(X), T(Y))$. Given the extension, the multiplication is given by $\mu_X = \mathrm{id}^\sharp_X$, and given the multiplication, the extension can be recovered by $f^\sharp(x) = \mu_X(T(f)(x))$. When $\mathcal{C}$ is monoidal, monads can also be equipped with a strength τ, a natural transformation such that for $X, Y \in \mathcal{C}$, $\tau_{X,Y} : X \otimes T(Y) \to T(X \otimes Y)$. Such monads are called *strong*.

Strong monads capture many common computational effects; throughout the paper we will refer to the writer monad $W(X) = X \times M$ for some fixed monoid M, the state monad $S(X) = S \to X \times S$ for a fixed state set S, the continuation passing style (CPS) monad $C(X) = (X \to \mathsf{Ans}) \to \mathsf{Ans}$ for a fixed set Ans, and the probabilistic powerdomain monad (see [24]). For further discussion on monads we refer the reader to Perrone [35].

3 Toy Language

Throughout the following section, we will develop a toy language for hybrid quantum-classical computation (introduced in Section 3.1), slowly adding various quantum features as effects. In parallel, we describe a denotational semantics for each language, which we slowly adapt to incorporate each feature, motivating the final definition of the quantum orchestra monad.

In Section 3.2, we add the ability to trigger the execution of unitary quantum gates on a fixed quantum space, creating a circuit description language whose terms can be understood as classical functions which print out a circuit. In this language, qubits are references by a fixed finite set of constants, which have predefined meaning in the quantum space we are manipulating.

In Section 3.3, we add measurement. As the program can utilise the (classical) outcomes of measurement, it no longer produces static circuits for each input, instead exhibiting non-trivial hybrid interactions. In this section we will show how each program can be modelled by a quantum instrument, crucially describing how these instruments can be composed.

Next, we add the ability to allocate fresh qubits in Section 3.4. This allows the size of our quantum state to vary throughout the computation, and requires the addition of non-constant qubit references. We semantically model this by moving to a parameterised monad in the style of Atkey [3], where computations are parameterised by their input and output quantum spaces.

Lastly, in Section 4 we move our model from sets to *directed-complete partial orders*, allowing the addition of fixpoints combinators to our toy language. In this section we finally give a full definition of our quantum orchestra monad, and the associated operations that can be performed on it.

3.1 A Pure Classical Base Language

All variants of our toy language are built on the same classical foundation. Our presentation closely follows Kammar et al. [25] (following [33, 34]), who use a similar language to study a computational effect. In particular we have types given by the following syntax:

$$X, Y \ ::= \ 1 \mid X \times Y \mid \mathsf{bool} \mid X \to Y$$

$$\frac{x : X \in \Gamma}{\Gamma \vdash^v x : X} \qquad \frac{}{\Gamma \vdash^v () : 1} \qquad \frac{\Gamma \vdash^v v_1 : X \qquad \Gamma \vdash^v v_2 : Y}{\Gamma \vdash^v (v_1, v_2) : X \times Y} \qquad \frac{\Gamma \vdash^v v : X \times Y}{\Gamma \vdash^v \mathsf{fst}\, v : X} \qquad \frac{\Gamma \vdash^v v : X \times Y}{\Gamma \vdash^v \mathsf{snd}\, v : Y}$$

$$\frac{}{\Gamma \vdash^v \mathsf{True} : \mathsf{bool}} \qquad \frac{}{\Gamma \vdash^v \mathsf{False} : \mathsf{bool}} \qquad \frac{\Gamma, x : X \vdash^e e : Y}{\Gamma \vdash^v \lambda x.e : X \to Y} \qquad \frac{\Gamma \vdash^v v : X}{\Gamma \vdash^e \mathsf{return}\, v : X}$$

$$\frac{\Gamma \vdash^e e_1 : X \qquad \Gamma, x : X \vdash^e e_2 : Y}{\Gamma \vdash^e \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2 : Y} \qquad \frac{\Gamma \vdash^e e_1 : X \qquad \Gamma \vdash^e e_2 : Y}{\Gamma \vdash^e e_1; e_2 : Y}$$

$$\frac{\Gamma \vdash^v v_1 : X \to Y \qquad \Gamma \vdash^v v_2 : X}{\Gamma \vdash^e v_1(v_2) : Y} \qquad \frac{\Gamma \vdash^v v : \mathsf{bool} \qquad \Gamma \vdash^e e_1 : X \qquad \Gamma \vdash^e e_2 : X}{\Gamma \vdash^e \mathsf{if}\, v \,\mathsf{then}\, e_1 \,\mathsf{else}\, e_2 : X}$$

Fig. 3. Typing rules for the classical fragment of the toy language

Although we foresee no difficulties with their inclusion, we omit more general sum types, including just a boolean type which is necessary to describe the classical outcomes of measurement.

The language is a form of fine-grain call-by-value lambda calculus [33], and as such we split its terms into *values*, terms which do not compute and can be bound to variables, and *expressions*, terms representing computations which may trigger effects. Values and expression are respectively given by the following grammars.

$$\begin{aligned} v &::= x \mid () \mid (v_1, v_2) \mid \mathsf{fst}\, v \mid \mathsf{snd}\, v \mid \mathsf{True} \mid \mathsf{False} \mid \lambda x.e \\ e &::= \mathsf{return}\, v \mid \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2 \mid e_1; e_2 \mid v_1(v_2) \mid \mathsf{if}\, v \,\mathsf{then}\, e_1 \,\mathsf{else}\, e_2 \end{aligned}$$

As with the types, we have not included all possible constructions, but just enough to exhibit non-trivial behaviour and interaction with our quantum effects. For example, we do not include a case matching/pattern matching construction for products, instead relying on projections. We do however include a "let" binding, even though it can be emulated with λ-abstraction and application, in order to increase the legibility of our example programs. We also include $e_1\,; e_2$, the sequential composition of expressions, as a simplification of the "let" binding for the case where e_2 does not require the value produced by e_1.

Our typing rules are split into two judgements: $\Gamma \vdash^v v : A$ for values and $\Gamma \vdash^e e : A$ for expressions. While they are standard, we include them in Fig. 3, as we build on them throughout the following sections.

We give semantics to this language in a standard way. First, a semantics $[\![X]\!]$ is given to each type X, which denotes the possible closed values of that type. The semantics of an (open) value $\Gamma \vdash^v v : X$ will then simply be a function $[\![v]\!] : [\![\Gamma]\!] \to [\![X]\!]$, where:

$$[\![x_1 : X_1, \ldots, x_n : X_n]\!] = [\![X_1]\!] \times \cdots \times [\![X_n]\!]$$

and we let $\rho_x \in [\![X]\!]$ be the component of x for $x : X \in \Gamma$ and $\rho \in [\![\Gamma]\!]$.

We could give semantics to expressions in the same way, but this would not extend to the effectful computations of the following sections. We instead suppose that we have an effect given by some strong monad T on a cartesian closed category C, and then the semantics of an expression $\Gamma \vdash^e e : X$ will be given by a function:

$$[\![e]\!] : [\![\Gamma]\!] \to T[\![X]\!]$$

Essentially, an expression of type X, produces a value of type X and effects governed by T. Of course, one could instantiate T as the identity monad to obtain a semantics for the pure language above. Equipped with this, we can now give semantics to each type:

$$[\![1]\!] = \{()\} \quad [\![X \times Y]\!] = [\![X]\!] \times [\![Y]\!] \quad [\![\mathsf{bool}]\!] = \{\mathsf{True}, \mathsf{False}\}$$

A (closed) value of type $X \to Y$ then represents a function that takes a *value* of type X, and returns an *expression* of type Y, and so should have semantics:

$$[\![X \to Y]\!] = [\![X]\!] \to T[\![Y]\!]$$

The semantics of values in context Γ is then standard:

$$\begin{aligned} [\![x]\!](\rho) &= \rho_x & [\![(v_1, v_2)]\!](\rho) &= ([\![v_1]\!](\rho), [\![v_2]\!](\rho)) \\ [\![()]\!](\rho) &= () & [\![\mathsf{fst}\, v]\!](\rho) &= \pi_1([\![v]\!](\rho)) \\ [\![\mathsf{True}]\!](\rho) &= \mathsf{True} & [\![\mathsf{snd}\, v]\!](\rho) &= \pi_2([\![v]\!](\rho)) \\ [\![\mathsf{False}]\!](\rho) &= \mathsf{False} & [\![\lambda x.e]\!](\rho)(\rho_x) &= [\![e]\!](\rho, \rho_x) \end{aligned}$$

where π_1 and π_2 are the projections from the product. The semantics of each expression is also standard and can be given in terms of the monad operations (where η is the unit, τ is the strength, and extension of f is given by $f^\sharp$).

$$[\![\mathsf{return}\, v]\!] = [\![\Gamma]\!] \xrightarrow{[\![v]\!]} [\![X]\!] \xrightarrow{\eta_X} T[\![X]\!]$$

$$[\![\mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2]\!] = [\![\Gamma]\!] \xrightarrow{\langle \mathrm{id}, [\![e_1]\!]\rangle} [\![\Gamma]\!] \times T[\![X]\!] \xrightarrow{\tau_{[\![\Gamma]\!],[\![X]\!]}} T([\![\Gamma]\!] \times [\![X]\!]) \xrightarrow{[\![e_2]\!]^\sharp} T[\![Y]\!]$$

$$[\![e_1; e_2]\!] = [\![\Gamma]\!] \xrightarrow{\langle \mathrm{id}, [\![e_1]\!]\rangle} [\![\Gamma]\!] \times T[\![X]\!] \xrightarrow{\tau_{[\![\Gamma]\!],[\![X]\!]}} T([\![\Gamma]\!] \times [\![X]\!]) \xrightarrow{T\pi_1} T[\![\Gamma]\!] \xrightarrow{[\![e_2]\!]^\sharp} T[\![Y]\!]$$

$$[\![v_1(v_2)]\!] = [\![\Gamma]\!] \xrightarrow{\langle [\![v_1]\!], [\![v_2]\!]\rangle} ([\![X]\!] \to T[\![Y]\!]) \times [\![X]\!] \to T[\![Y]\!]$$

$$[\![\mathsf{if}\, v \,\mathsf{then}\, e_1 \,\mathsf{else}\, e_2]\!] = [\![\Gamma]\!] \xrightarrow{\langle [\![v]\!], \mathrm{id}\rangle} [\![\mathsf{bool}]\!] \times [\![\Gamma]\!] \simeq [\![\Gamma]\!] + [\![\Gamma]\!] \xrightarrow{[[\![e_1]\!], [\![e_2]\!]]} T[\![X]\!]$$

where $\langle f, g\rangle$ is the universal map into the product, and $[f, g]$ is the universal map out of the sum.

3.2 A Classical Language with a Quantum Effect

We begin our investigation of quantum effects by augmenting our classical language with a side-effect for performing unitary quantum operations, allowing it to describe (parameterised) circuits. As is common for describing pure quantum computations, we reduce the problem of describing all possible unitaries we could want to describing a fixed *gate set*. To describe which qubits each gate is applied to, we add a type qid of *qubit references*:

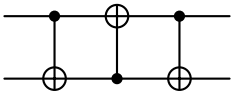
$$X, Y \ ::= \ \boxed{\mathsf{qid}} \mid 1 \mid X \times Y \mid \mathsf{bool} \mid X \to Y$$

Our quantum gates are then simply functions which take some number of qubit references as input (corresponding to the number of qubits the gate acts on) and return unit. To introduce terms of type qid, we add a set of constants $l \in \mathbf{Q}$ to the language, which index a fixed quantum state. These constants have the following typing rules:

$$\frac{q \in \mathbf{Q}}{\Gamma \vdash^v q : \mathsf{qid}} \qquad \frac{\mathrm{G} \in \{\mathrm{H}, \mathrm{S}, \mathrm{X}, \mathrm{T}\}}{\Gamma \vdash^v \mathrm{G} : \mathsf{qid} \to 1} \qquad \frac{}{\Gamma \vdash^v \mathrm{CX} : \mathsf{qid} \times \mathsf{qid} \to 1}$$

We emphasise that gates do not compute without being applied to their qubit arguments, so all appear as values. When they are applied to a sequence of qubit references, these operations are understood to mutate the quantum state referenced by those variables as a side effect. In this presentation we have chosen to use the universal Clifford+T gate set, but this could be substituted by any reasonable alternative, including those parameterised by booleans (classically conditioned gates) or angles (e.g. RZ gates), though doing so would require the addition of angle types to the language, which we avoid for simplicity.

Example 15 (Swap gate)**.** Consider the following circuit implementing the swap gate:



This can be naturally implemented as a function in this toy language:

$$\vdash^{v} \lambda q.\mathrm{CX}((\mathsf{fst}\, q, \mathsf{snd}\, q)); \mathrm{CX}((\mathsf{snd}\, q, \mathsf{fst}\, q)); \mathrm{CX}((\mathsf{fst}\, q, \mathsf{snd}\, q)) : \mathsf{qid} \times \mathsf{qid} \to 1$$

Such a function could be applied in the same way as a primitive gate.

3.2.1 A Unitary Effect. To extend the semantics of Section 3.1 to accommodate the new features of this section, we must instantiate the monad T describing the effect, and give an interpretation of each new type and term constructor. At this stage, all our effect can do is generate some unitary evolution as the program progresses. We will assume that that these unitaries act on the space $\mathcal{H}$ consisting of exactly the qubits present in $\mathbf{Q}$:

$$\mathcal{H} = \bigotimes_{\mathbf{Q}} \mathbb{C}^2$$

For a choice of qubit $q \in \mathbf{Q}$, each single qubit quantum operation $U \in \{\mathrm{H}, \mathrm{X}, \mathrm{S}, \mathrm{T}\}$ then induces a unitary U_q which applies the gate to the given qubit. Pre-empting the following sections, we instead associate to the quantum operation the map:

$$\overline{U_q}(M) = U_q^\dagger M U_q$$

which is a complete positive unital map from $\mathcal{B}(\mathcal{H})$ to itself. The interpretation of CX raises an interesting question. If the two qubit locations passed to the operations are distinct, then we can proceed as before, letting $\overline{\mathrm{CX}_{q_1,q_2}}$ be the appropriate map on $\mathcal{B}(\mathcal{H})$, but what should we when the qubit references are the same, that is what do we assign $\overline{\mathrm{CX}_{q,q}}$? One option is to enforce that this can never occur at the type system level (as done in Guppy [31]). In this paper, we can instead "fail" at runtime by letting:

$$\overline{\mathrm{CX}_{q,q}} = 0_{\mathcal{B}(\mathcal{H})}$$

Such a map is not unital, yet is still subunital, and so we are able to assign a subunital operation to every quantum operation for every choice of arguments.

An expression $\Gamma \vdash^{e} e : X$ produces some value of type X, and also applies some evolution to the quantum space, and so its semantics is the function:

$$[\![e]\!] : [\![\Gamma]\!] \to [\![X]\!] \times \mathbf{W}^*(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}))$$

And so by comparing the with the denotation of expressions in Section 3.1, we can see that the monad T should be given by the functor $\_ \times \mathbf{W}^*(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}))$, which is the writer/printing monad (see e.g. [35, Example 5.1.14]) with the monoid structure on $\mathbf{W}^*(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}))$ being given by composition and the identity map.

At this stage, with the restricted form of our qubit references, we can simply set $[\![\mathsf{qid}]\!] = \mathbf{Q}$, so that closed qubit references is given by one of the constants. Concretely, we have $[\![q]\!](\rho) = q$. All that remains is define the semantics of each gate, which is given below:

$$[\![G]\!](\rho)(q) = ((), \overline{G}_q) \text{ for } G \in \{\mathrm{H}, \mathrm{X}, \mathrm{S}, \mathrm{T}\} \quad [\![\mathrm{CX}]\!](\rho)((q_1, q_2)) = ((), \overline{\mathrm{CX}}_{q_1,q_2})$$

where each of these is a function $[\![\Gamma]\!] \to \mathbf{Q}^i \to \{()\} \times \mathbf{W}^*(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}))$ where $i = 2$ for CX and 1 otherwise, which we derive by expanding the semantics of types and the definition of the monad.

Example 16 (Classically Controlled Unitaries)**.** A classically controlled X gate can be given as the following function, where $q \in \mathbf{Q}$:

$$\textsc{condX} \coloneqq \lambda b.\, \mathsf{if}\ b\ \mathsf{then}\ \mathrm{X}(q)\ \mathsf{else}\ \mathsf{return}\ ()$$

As a closed term, this function has a denotation of the form:

$$[\![\textsc{condX}]\!] : \mathsf{bool} \to \{()\} \times \mathbf{W}^*(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}))$$

and a calculation shows that True is sent to $((), \overline{X_q})$ and False is sent to $((), \mathrm{id})$.

3.3 Instrumenting Effects Between a Classical and Quantum Computer

To achieve more interesting hybrid interaction in a quantum programming language, we must be able to perform quantum measurements and be able to interact with their classical output. To do so, we extend our toy language with an additional function MEAS that takes a qid and returns a bool:

$$\frac{}{\Gamma \vdash^v \textsc{meas} : \mathsf{qid} \to \mathsf{bool}}$$

Here, MEAS implements measurement of a single qubit in the computational basis. Computational basis measurement has two possible outcomes, $|0\rangle$ and $|1\rangle$ to which we associate to the classical outcomes $\{\mathsf{False}, \mathsf{True}\} = \mathsf{bool}$. Given that our previous semantics already utilised quantum channels, it may seem like we already have the tools necessary to model this measurement operation. Unfortunately, while the previous semantics can model the action of measurement on our quantum space, it does not model the classical outcome obtained. We recall the reset example from the introduction.

Example 17 (Reset)**.** The reset channel maps all states to $|0\rangle$. As in the introduction, this can be performed by first measuring a qubit, and then applying an X gate conditioned on the measurement result. This is realised by the following program, where we let $q \in \mathbf{Q}$ as in Example 16:

$$\textsc{reset} \coloneqq \mathsf{let}\ b = \textsc{meas}(q)\ \mathsf{in}\ \textsc{condX}(b)$$

Crucially, the outcome of the measurement is used as an argument to the classical conditioned X gate, and so the semantics of this channel cannot be realised by simply composing channels.

To model this action we move from quantum channels to *quantum instruments*, as introduced in Section 2.3. A quantum instrument assigns a quantum channel to each potential classical output, with the each component giving the evolution of the quantum state if we had postselected for that classical output, and the subnormalisation factor giving the probability of receiving that output. More concretely, for the computational basis measurement we have the instrument:

$$\delta(\mathsf{False}, \overline{|0\rangle\langle 0|}) + \delta(\mathsf{True}, \overline{|1\rangle\langle 1|})$$

where δ is the Dirac quantum instrument and we recall that for a linear map V, we define the channel $\overline{V}(M) = V^\dagger M V$. We note that the combination of the components is subunital as required:

$$(\overline{|0\rangle\langle 0|} + \overline{|1\rangle\langle 1|})(1) = |0\rangle\langle 0|\, 1\, |0\rangle\langle 0| + |1\rangle\langle 1|\, 1\, |1\rangle\langle 1| = |0\rangle\langle 0| + |1\rangle\langle 1| = 1$$

such an instrument is exactly the tool we need to give semantics to our language with measurement, as it captures both the quantum effect and the classical outcome of the operation. We will therefore define the semantics of this language by letting the monad T be given by Q, where $Q(X)$ is the set of quantum instruments on our fixed quantum space $\mathcal{H}$ with classical outcomes X. We then define:

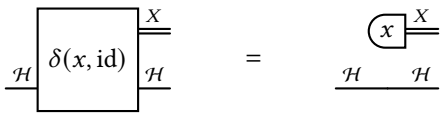
$$[\![\textsc{meas}]\!](\rho)(q) = \delta(\mathsf{False}, \overline{|0\rangle\langle 0|_q}) + \delta(\mathsf{True}, \overline{|1\rangle\langle 1|_q})$$

where as in the previous section, the subscripted q refers to map which acts as indicated on the given qubit, and acts as the identity on all other qubits. We must also be careful that our quantum operations from the previous section still have an interpretation in the new monad, but this is immediate from the following.

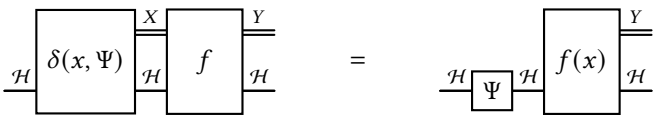
Remark 18. A quantum instrument on the set $\{()\}$ is precisely a quantum channel: every channel Ψ gives rise to an instrument $\delta((), \Psi)$, and every instrument on $\{()\}$ is of this form.

3.3.1 Composing Quantum Instruments. All that remains to complete our semantics is to show that Q is actually a monad. Critically, to make sense of programs whose effects are governed by quantum instruments, we must define what it means to compose these instruments. Recalling Fig. 1, the required composition is the monadic composition depicted in Fig. 1c, where an instrument $\xi \in Q(X)$ is composed with a Kleisli map $f : X \to Q(Y)$. We defer a full description of the monad to Appendix B, partially as it a stepping stone to the full quantum orchestra monad given in Section 4. Here we instead describe the action of each monad operation on Dirac quantum instruments, which is sufficient for building intuition.

The unit of the monad is already represented as a Dirac instrument, with $\eta_X(x) = \delta(x, \mathrm{id}_{\mathcal{B}(\mathcal{H})})$. Intuitively, this instrument does not effect the quantum state, and could be drawn as:

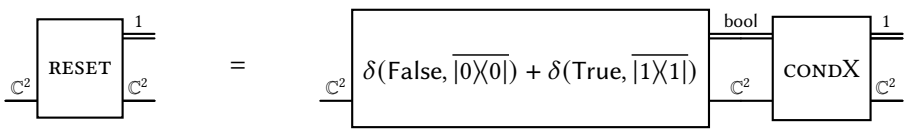


Now we consider the monadic composition of a Dirac instrument $\delta(x, \Psi)$ and a Kleisli map $f : X \to Q(Y)$, given by $f^\sharp(\delta(x, \Psi))$. We can again depict this graphically:



and we see that the y component of $f^\sharp(\delta(x, \Psi))$ is given the y component of $f(x)$ composed with Ψ (recalling we work in the Heisenberg picture). The full definition of the monad in Appendix B implies that this action can be extended by linearity to all finite quantum instruments, despite Dirac decompositions not being unique.

With the definition of the monad, and the semantics of the new constructor, we have a full description of the semantics of this iteration of the toy language. We can now return to the reset example and calculate its semantics. For simplicity, we will assume we work in the single qubit space, that is $\mathbf{Q} = \{q\}$. We are then in the following situation:



We can now calculate the component of each side for $() \in [\![1]\!]$:

$$\begin{aligned}
[\![\textsc{reset}]\!] &= [\![\textsc{condX}]\!]^\sharp([\![\textsc{meas}]\!]) \\
&= [\![\textsc{condX}]\!]^\sharp(\delta(\mathsf{False}, \overline{|0\rangle\langle 0|}) + \delta(\mathsf{True}, \overline{|1\rangle\langle 1|})) \\
&= [\![\textsc{condX}]\!]^\sharp(\delta(\mathsf{False}, \overline{|0\rangle\langle 0|})) + [\![\textsc{condX}]\!]^\sharp(\delta(\mathsf{True}, \overline{|1\rangle\langle 1|})) \\
&= \overline{|0\rangle\langle 0|} \circ [\![\textsc{condX}]\!](\mathsf{False}) + \overline{|1\rangle\langle 1|} \circ [\![\textsc{condX}]\!](\mathsf{True}) \\
&= \overline{|0\rangle\langle 0|} \circ \mathrm{id} + \overline{|1\rangle\langle 1|} \circ \overline{X} \\
&= \overline{|0\rangle\langle 0|} + \overline{|0\rangle\langle 1|}
\end{aligned}$$

where we have omitted the arguments for the semantics of the empty context and for taking the appropriate component, as both are singleton sets. From this is it clear that our program for reset is semantically identical to the true reset channel in the Heisenberg picture we are working within.

3.4 Allocating Qubits

So far, our programs have all operated on a fixed quantum space governed by a set of constants $\mathbf{Q}$. While this aligns closely with hardware, it forces the programmer to calculate how much memory is needed, or dually, to know how much memory is available on their target hardware. Ideally, the programmer should be alleviated of this concern, and be able to request new memory when needed through a new function $\textsc{alloc} : 1 \to \mathsf{qid}$.

To accommodate this in our semantics, the quantum space our instruments act on must be continually updated throughout the computation. Further, it should be possible for quantum instruments to have different input and output spaces. We achieve this by generalising our construction to a *parameterised monad* (see Atkey [3]), $Q(\mathscr{A}, \mathscr{B}, X)$, where X is the set acted on by the monad, with $\mathscr{A}$ and $\mathscr{B}$ being the input and output W*-algebras. The monad operations work similarly to before, with the constraint that the input and output spaces must align when appropriate. We include full details of the parameterised construction for finite instruments in Appendix B.

Now that we are able to paramaterise our effect, the natural question is where to get the input and output parameters from. To address the second of these questions, we adapt our typing to track the number of qubits allocated by an expression using a grading, reminiscent of graded type-and-effect systems [16, 26, 27]. The function type is replaced with a graded set of function types:

$$X \to_n Y$$

and the judgement for expressions is given by $\Gamma \vdash^e_n e : X$. The updated typing rules to accommodate these grades are standard, and are given in full in Appendix A. We highlight the rule for the if statement, and the new rule for $\textsc{alloc}$:

$$\frac{\Gamma \vdash^v v : \mathsf{bool} \qquad \Gamma \vdash^e_n e_1 : X \qquad \Gamma \vdash^e_n e_2 : X}{\Gamma \vdash^e_n \mathsf{if}\, v\, \mathsf{then}\, e_1\, \mathsf{else}\, e_2 : X} \qquad\qquad \frac{}{\Gamma \vdash^v \textsc{alloc} : 1 \to_1 \mathsf{qid}}$$

The $\textsc{alloc}$ function allocates exactly one qubit and returns a references to this new qubit, which is tracked in its type. The rule for the if statement enforces that both expressions must allocate the same number of qubits, ensuring that the total allocation is static. This allows us to determine how much larger the output parameter should be than the input.

Determining the input space is complicated by the fact that an expression could be run in a variety of quantum spaces, sometimes within the same program. To remedy this, we define the semantics of a term in all possible quantum spaces, much like how the semantics of a term is given

in semantic contexts. The semantics of an expression is then given by a function:

$$[\![e]\!]_m \in \mathbf{DCPO}([\![\Gamma]\!]_m, \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n}), [\![X]\!]_{m+n}))$$

and is defined for every number of starting qubits m, with the final space having $m + n$ qubits. We have chosen here to also parametrise the semantics of types over natural numbers, allowing us to define:

$$[\![\mathsf{qid}]\!]_m = \{0, \ldots, m-1\}$$

ensuring that qubit references are always "in scope". To define the semantics of "let", we now require an inclusion function $\iota_{\Gamma,m,m'} : [\![\Gamma]\!]_m \to [\![\Gamma]\!]_{m'}$ for $m' \geq m$. Doing is not entirely trivial, requiring the semantics of types to be given as:

$$[\![X \to_n Y]\!]_m = \forall m' \geq m. \mathbf{DCPO}([\![X]\!]_{m'}, \mathcal{Q}(\mathcal{B}(\mathcal{H}^{m'}), \mathcal{B}(\mathcal{H}^{m'+n}), [\![Y]\!]_{m'+n}))$$

ensuring that $[\![X \to_n Y]\!]_m \subseteq [\![X \to_n Y]\!]_{m'}$ for $m' \geq m$. Finally, we have:

$$[\![\textsc{alloc}]\!]_m(\rho)(u) = \delta(m, \overline{V}) \quad \text{where } V(v) = v \otimes |0\rangle$$

recalling that $\overline{V}(M) = V^\dagger M V$. We emphasise that the action of ALLOC depends on the size of the input quantum space, both for the quantum channel and the returned qubit reference.

Example 19 (Bell pair allocation)**.** The following expression $\vdash^e_2 \mathsf{GHZ} : \mathsf{qid} \times \mathsf{qid}$ allocates a (two-qubit) Bell pair:

$$\textsc{Bell} \coloneqq \mathsf{let}\ q_1 = \textsc{alloc}(())\ \mathsf{in}\{\mathrm{H}(q_1); \mathsf{let}\ q_2 = \textsc{alloc}(())\ \mathsf{in}\{\mathrm{CX}(q_1, q_2); \mathsf{return}(q_1, q_2)\}\}$$

The returned qubits references depend on the size of the quantum space this program is executed in. In general we have:

$$[\![\textsc{Bell}]\!]_m = \delta((m, m+1), \overline{\phi}) \in \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+2}), \{0, \ldots, m+1\}^2)$$

where $\phi(v) = v \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. We note that the inclusion $\iota_{\mathsf{qid},m,m+1}$ is used to include the returned qubit reference from the first ALLOC into the set $\{0, \ldots, m+1\}$.

This restricted form of static allocation through graded types is of course not the only form that allocation could take, though we choose the form as we believe it best showcases the parameterised nature of the monad. An alternative to our ALLOC function could be a construction:

$$\frac{\Gamma, a : \mathsf{qid} \vdash^e e : X}{\Gamma \vdash^e \mathsf{ancilla}\, a\, \mathsf{in}\, e : X}$$

which allocates an ancilla qubit, executes the expression e in the larger space, and releases the qubit afterwards. Such behaviour could also be captured by our parameterised monad (though this would only ever need invocations of the monad where the input and output space are the same).

Finally, a more flexible allocation method could be modelled by fixing a single infinite dimensional W*-algebra instead of parametrising, utilising that the monad is defined for all W*-algebras, and not just the finite dimensional ones we have used so far. We believe that with the appropriate algebra, the monad could support the allocation operation above without the need to enforce that it is only used statically, allowing it to appear in conditional statements, and even loops (see Section 4).

4 Modelling Diverging Quantum Programs

To model programs with both quantum effects and divergence, we extend our finite instrument monad of the previous sections to a "Quantum Orchestra" monad $\mathcal{Q}$ on the category **DCPO**. We motivate this by investigating a *repeat-until-success* circuit [1].

Example 20 (Repeat-Until-Success)**.** Let U be some unitary we want to perform on $\mathcal{H}$, and suppose we have a (closed) computation $\vdash^e C : \mathsf{bool}$ with a single classical boolean output such that:

- if the classical output is False, then the quantum state of $\mathcal{H}$ remains unchanged,
- if it is True, then U is performed on the state.

A repeat-until-success circuit performs the evolution U by repeatedly running the computation C until a True outcome is observed. We can model this in our setting as follows: the semantics of C can be given by the following instrument, where p is a fixed probability of observing True:

$$[\![C]\!] : \mathcal{Q}(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}), \mathbf{2})$$

$$[\![C]\!] = \delta(\mathsf{True}, p\overline{U}) + \delta(\mathsf{False}, (1-p)\,\mathrm{id}_{\mathcal{B}(\mathcal{H})})$$

We then define the repeat until success circuit recursively by:

$$\mathrm{RUS} \coloneqq \mathsf{let}\ b = C\ \mathsf{in}\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{return}\ ()\ \mathsf{else}\ \mathrm{RUS}$$

A standard method to solve this recursive equation is through a fixpoint:

$$\mathrm{RUS} \coloneqq \mathsf{fix}\ \lambda r.\ \mathsf{let}\ b = C\ \mathsf{in}\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{return}\ ()\ \mathsf{else}\ r(())$$

where $r : 1 \to 1$. Using our instrument monad, we could calculate (up to the isomorphism):

$$[\![\lambda r.\ \mathsf{let}\ b = C\ \mathsf{in}\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{return}\ ()\ \mathsf{else}\ r(())]\!] : \mathcal{Q}(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}), 1) \to \mathcal{Q}(\mathcal{B}(\mathcal{H}), \mathcal{B}(\mathcal{H}), 1)$$

but could not in general take a fixpoint of this function. By extending the monad to **DCPO**, assuming the function above were monotone we would be able to define:

$$[\![\mathrm{RUS}]\!] = \mathsf{fix}\,[\![\lambda r.\ \mathsf{let}\ b = C\ \mathsf{in}\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{return}\ ()\ \mathsf{else}\ r(())]\!]$$

where it could then be proved that this program has the desired properties.

An immediate consequence of modelling divergence is that the finiteness condition on our instruments is too restrictive: if we adapted the computation of Example 20 to return a counter of how many iterations occurred, then the resulting instrument would clearly have non-zero components for each natural number. This complicates taking the summations used to define the monad multiplication. A second observation is that an appropriate ordering on instruments cannot be given pointwise on components. Despite this ordering making the set of quantum instruments a dcpo, it is too coarse: the unit of the monad is not continuous in this ordering as $x < y$ would imply that $\eta(x)$ is incomparible to $\eta(y)$.

A potential solution to both problems, inspired by the probabilistic powerdomain monad [24], is for instruments to have a component for each Scott-open subset (those that are upwards closed and inaccessible from below), as opposed to each element, subject to the conditions of being a *valuation*. Under this interpretation, the multiplication must then be defined by an integral construction, and it must be shown that this integral construction is monotone. The techniques used in the classical (probabilistic) case do not neatly transfer to our quantum case, and we will compare these approaches in Section 6.

Instead of using valuations, we instead extend further to a continuation monad of "Quantum orchestras", where instead of having a component for each open set of a dcpo X, we have a component for each continuous function $X \to \mathbf{W}^*(\mathscr{B}, \mathscr{A})$. Orchestras restrict to valuations, by considering the components associated to indicator functions. Intuitively, the data of an orchestra

is that of a valuation and a definition of integral for that valuation. Crucially, this allows the order to be defined in a way that makes this bundled integration function continuous by definition.

We again define this monad as a $\mathbf{W}^{*\mathrm{op}}$-parameterised monad. We note that the resulting form of the monad is very close to the continuation passing style monad, from which we inherit much of the monad structure.

Definition 21 (Parameterised Quantum Orchestra Monad)**.** We define the $\mathbf{W}^{*\mathrm{op}}$-parameterised quantum orchestra monad $\mathcal{Q}$ as follows:

- Let $\mathscr{A}, \mathscr{B}$ be W*-algebras and X be a **DCPO**. Then elements $\xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ are given by a collection:
$$\{\xi_k \in \mathbf{W}^*(\mathscr{K}, \mathscr{A}) \mid \mathscr{K} \in \mathbf{W}^*, k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))\}$$
satisfying:
  - **Continuity:** For each $\mathscr{K}$ and $\mathscr{C}$, the function
$$\lambda k.\xi_k : \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B})) \to \mathbf{W}^*(\mathscr{K}, \mathscr{A})$$
is Scott-continuous.
  - **Subconvexity:** ξ has a restricted form of linearity. Given $\theta, \rho \in \mathbb{R}^+$ with $\theta + \rho \leq 1$ we have for all $k, k' \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$:
$$\xi_{\lambda x.\theta k(x) + \rho k'(x)} = \theta \xi_k + \rho \xi_{k'}$$
  - **Compositionality:** If $\chi \in \mathbf{W}^*(\mathscr{K}', \mathscr{K})$, then:
$$\xi_{\lambda x.k(x) \circ \chi} = \xi_k \circ \chi$$
for all $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$.

  We define the ordering pointwise: $\xi \leq \xi'$ if $\xi_k \leq \xi'_k$ for all k, where the ordering on $\mathbf{W}^*(\mathscr{K}, \mathscr{B})$ is given by the standard Löwner order. For a directed set $\Delta \subseteq \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$, we define its supremum by:
$$(\sup \Delta)_k = \sup\{\xi_k \mid \xi \in \Delta\}$$
for all $\mathscr{K}$, and $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$, using that $\mathbf{W}^*$ is **DCPO**-enriched (see Theorem 10).
- The action of the functor $\mathcal{Q}$ on $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}', \mathscr{B})$, and $f \in \mathbf{DCPO}(X, Y)$ is given by:
$$\mathcal{Q}(\phi, \psi, f)(\xi)_k = \phi \circ \xi_{\lambda x.\psi \circ k(f(x))}$$
where $k \in \mathbf{DCPO}(Y, \mathbf{W}^*(\mathscr{K}, \mathscr{B}'))$ and $\xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$.
- The unit $\eta_{\mathscr{A}X} : X \to \mathcal{Q}(\mathscr{A}, \mathscr{A}, X)$ is given by:
$$\eta_{\mathscr{A}X}(x)_k = k(x)$$
for $x \in X$ and $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{A}))$.
- The multiplication $\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X} : \mathcal{Q}(\mathscr{A}, \mathscr{B}, \mathcal{Q}(\mathscr{B}, \mathscr{C}, X)) \to \mathcal{Q}(\mathscr{A}, \mathscr{C}, X)$ is given by:
$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_k = \Xi_{\lambda \xi.\xi_k}$$
for $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{C} \otimes \mathscr{E}))$.
- The strength $\tau_{X,\mathscr{A},\mathscr{B},Y} : X \times \mathcal{Q}(\mathscr{A}, \mathscr{B}, Y) \to \mathcal{Q}(\mathscr{A}, \mathscr{B}, X \times Y)$ is defined by:
$$\tau_{X,\mathscr{A},\mathscr{B},Y}(x, \xi)_k = \xi_{\lambda y.k(x,y)}$$
for $k \in \mathbf{DCPO}(X \times Y, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$.

In Section 5, we will prove that the definition above actually forms a monad, as we have claimed. We next define the *Dirac* quantum orchestra, the analogue of the Dirac quantum instrument. As each of the quantum constructors in Sections 3.3 and 3.4 were given in terms of Dirac instruments, this construction allows us to immediately transfer their definition to the orchestra monad.

Definition 22 (Dirac quantum orchestra)**.** Given $x \in X$ and $\Phi \in \mathbf{W}^*(\mathscr{B}, \mathscr{A})$, we can define the *Dirac quantum orchestra* $\delta(x, \Phi) \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ by:

$$\delta(x, \Phi)_k = \Phi \circ k(x)$$

for all $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$. To show this is indeed a quantum orchestra, we first note that continuity and subconvexity follow from suprema and subconvex combinations being defined pointwise on the space $\mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$, and from the composition of morphisms in $\mathbf{W}^*$ being continuous and linear. For compositionality, we have:

$$\delta(x, \Phi)_{\lambda x.k(x)\circ\chi} = \Phi \circ k(x) \circ \chi = \delta(x, \Phi)_k \circ \chi$$

for $\chi \in \mathbf{W}^*(\mathscr{K}', \mathscr{K})$ and $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{A}))$, as required.

We pause to consider the action of each of the monad operations on Dirac quantum orchestras. Although it does not take an orchestra as an argument, it is immediate that the unit $\eta_{\mathscr{A}X}(x)$ is given by the Dirac quantum orchestra $\delta(x, \mathrm{id}_{\mathscr{A}})$. Given $x \in X$, $\Phi \in \mathbf{W}^*(\mathscr{B}, \mathscr{A})$, $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}', \mathscr{B})$, and $f \in \mathbf{DCPO}(X, Y)$, we have:

$$\mathcal{Q}(\phi, \psi, f)(\delta(x, \Phi))_k = \phi \circ \Phi \circ \psi \circ k(f(x)) = \delta(f(x), \phi \circ \Phi \circ \psi)_k$$

for all $k \in \mathbf{DCPO}(Y, \mathbf{W}^*(\mathscr{K}, \mathscr{B}'))$. For the multiplication, consider $\xi \in \mathcal{Q}(\mathscr{B}, \mathscr{C}, X)$ and $\Phi \in \mathbf{W}^*(\mathscr{B}, \mathscr{A})$. Then for $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{C}))$ we have:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\delta(\xi, \Phi))_k = \delta(\xi, \Phi)_{\lambda\xi'.\xi'_k} = \Phi \circ \xi_k = \mathcal{Q}(\Phi, \mathscr{C}, X)(\xi)_k$$

We consider applying the strength to $\delta(y, \Phi)$ where $y \in Y$ and $\Phi \in \mathbf{W}^*(\mathscr{B}, \mathscr{A})$. For $x \in X$ and $k \in \mathbf{DCPO}(X \times Y, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}(x, \delta(y, \chi))_k = \delta(y, \chi)_{\lambda y', k(x,y')} = (\chi \otimes \mathscr{C}) \circ k(x, y) = \delta((x, y), \chi)_k$$

Lastly, we consider the extension operation for this monad. We gave the monad multiplication above to align with the known monad laws for parameterised monads, but can recover the extension of $f : X \to \mathcal{Q}(\mathscr{B}, \mathscr{C}, Y)$ as:

$$f^\sharp : \mathcal{Q}(\mathscr{A}, \mathscr{B}, X) \to \mathcal{Q}(\mathscr{A}, \mathscr{C}, Y) \qquad f^\sharp(\xi) = \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\mathcal{Q}(\mathscr{A}, \mathscr{B}, f)(\xi))$$

applying this to a Dirac quantum orchestra $\delta(x, \Psi) \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$, we get:

$$f^\sharp(\delta(x, \Psi)) = \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\mathcal{Q}(\mathscr{A}, \mathscr{B}, f)(\delta(x, \Psi))) = \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\delta(f(x), \Psi)) = \mathcal{Q}(\Psi, \mathscr{C}, Y)(f(x))$$

for a continuation $k \in \mathbf{DCPO}(Y, W^*(\mathscr{K}, \mathscr{C}))$, we further calculate:

$$f^\sharp(\delta(x, \Psi))_k = \mathcal{Q}(\Psi, \mathscr{C}, Y)(f(x))_k = \Psi \circ f(x)_k$$

We note the action of this monad on Dirac orchestras is entirely analogous to the action of the finite quantum instrument monad on Dirac instruments, and therefore the examples and reasoning from Section 3.3 embed into this setting. Much of this reasoning extends linearly to *finite quantum orchestras*, finite sums of Dirac orchestras, though characterising the order on finite orchestras is not trivial. We discuss finite orchestras in greater detail in Appendix C.

We can finally now define the final version of our toy language. The only addition over the language of Section 3.4 is the fix point operator, which has the following typing rule:

$$\frac{\Gamma \vdash^{v} f : (1 \to_0 X) \to_0 X}{\Gamma \vdash^{e}_0 \mathsf{fix}\, f : X}$$

We may have expected to apply a fixed point to an endofunction $f : X \to X$, but we note that $[\![X \to X]\!] = [\![X]\!] \to Q([\![X]\!])$ which is not suitable for taking a fixed point. If we had tried to remedy this by applying the extension to this function, the resulting endofunction would be strict, and hence its fixpoint would always be zero. In contrast:

$$[\![(1 \to X) \to X]\!] = (1 \to Q(X)) \to Q(X) \simeq Q(X) \to Q(X)$$

and so is suitable for taking a fixpoint. We note that this form of the fixpoint operator is reminiscent of the construction in Levy [32, Chapter 4], but with thunking replaced by functions from the unit.

For this language, we enforce at the type level that the function f performs no allocations, so that we can statically know the output space of the computation. As the monad can operate on infinite W*-algebras, we believe this is not a fundamental limitation, but we leave allocating within a loop to future work. Recalling Theorem 9, it is now routine to give semantics to this new constructor, which should take the form of a (continuous) function:

$$[\![\mathsf{fix}\, f]\!]_m \in \mathbf{DCPO}([\![\Gamma]\!], Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m))$$

and is given by the following composition:

$$\begin{aligned}[\![\mathsf{fix}\, f]\!]_m = [\![\Gamma]\!]_m &\xrightarrow{[\![f]\!]_m} \mathbf{DCPO}((1 \to Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)), Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)) \\ &\simeq \mathbf{DCPO}(Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m), Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)) \\ &\xrightarrow{\mathsf{fix}} Q(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)\end{aligned}$$

Crucially, the set $Q(\mathscr{A}, \mathscr{B}, X)$ is a pointed dcpo for any $\mathscr{A}$, $\mathscr{B}$, and X, with the bottom element being given by the the orchestra which maps every continuation to the zero channel, allowing us to take a fixpoint. The semantics of all other terms can be defined analogously to before, but for completeness we include a full description in Appendix A.

We end this section by recalling the repeat-until-success program RUS, and computing its meaning. Let us assume that $\Gamma \vdash^{e}_0 C : \mathsf{bool}$ and for a given $\rho \in [\![\Gamma]\!]_m$ we have:

$$[\![C]\!]_m(\rho) = \delta(\mathsf{True}, p\overline{U}) + \delta(\mathsf{False}, (1-p)\,\mathrm{id}_{\mathcal{B}(\mathcal{H})^m})$$

for some unitary U on $\mathcal{H}^m$ and $0 < p < 1$. Then, letting:

$$f \coloneqq \lambda r.\, \mathsf{let}\, b = C\, \mathsf{in}\, \mathsf{if}\, b\, \mathsf{then}\, \mathsf{return}() \,\mathsf{else}\, r(()) \qquad \mathrm{RUS} \coloneqq \mathsf{fix}\, f$$

we have (again up to the iso $(1 \to X) \simeq X$) by calculation that:

$$[\![f]\!]_m(\rho)(\xi) = \delta((), p\overline{U}) + (1-p)\xi$$

Now, it is clear that $([\![f]\!]_m(\rho))^n(\bot) = \delta((), (1-(1-p)^n)\overline{U})$, and so:

$$\mathsf{fix}([\![f]\!]_m(\rho)) = \sup_n([\![f]\!]_m(\rho))^n(\bot) = \sup_n \delta((), (1-(1-p)^n)\overline{U}) = \delta((), \overline{U})$$

as desired.

5 Quantum orchestras form a strong parameterised monad

We prove that the quantum orchestra construction given in Definition 21 is well defined, leading to Theorem 33 which states that $\mathcal{Q}$ is a strong $\mathbf{W}^{*\mathrm{op}}$-parameterised monad on **DCPO** with unit η, multiplication μ, and strength τ. We begin by showing that the action on objects produces dcpos.

Lemma 23. *$\mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ is a dcpo.*

Proof. As the ordering on $\mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ is pointwise on components, it is clear that it is a partial order. To show the suprema are well defined, fix a directed set $\Delta \subseteq \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$. We first show $\sup \Delta \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$:

- **Continuity**: Fix $\mathscr{K}$. Then the function:
$$\lambda k.(\sup \Delta)_k : \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B})) \to \mathbf{W}^*(\mathscr{K}, \mathscr{A})$$
is a supremum of functions $\{\lambda k.\xi_k \mid \xi \in \Delta\}$, each of which is continuous by the continuity of ξ. Hence $\lambda k.(\sup \Delta)_k$ is a supremum of continuous functions, and so is continuous.
- **Subconvexity**: Let $\theta, \rho \in \mathbb{R}^+$ with $\theta + \rho \leq 1$, and suppose $k, k' \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$. Then:
$$\begin{aligned}(\sup \Delta)_{\lambda x.\theta k(x) + \rho k'(x)} &= \sup\{\xi_{\lambda x.\theta k(x) + \rho k'(x)} \mid \xi \in \Delta\} \\ &= \sup\{\theta \xi_k + \rho \xi_{k'} \mid \xi \in \Delta\} \\ &= \theta \sup\{\xi_k \mid \xi \in \Delta\} + \rho \sup\{\xi_{k'} \mid \xi \in \Delta\} \\ &= \theta (\sup \Delta)_k + \rho (\sup \Delta)_{k'}\end{aligned}$$
Where scalar multiplication by a positive number is continuous because scalar multiplication by a positive number is a poset isomorphism. Similarly, adding a fixed channel is a poset isomorphism, and so addition is continuous as it is separately continuous in both arguments.
- **Compositionality**: Suppose $\chi \in \mathbf{W}^*(\mathscr{K}', \mathscr{K})$. Then:
$$\begin{aligned}(\sup \Delta)_{\lambda x.k(x) \circ \chi} &= \sup\{\xi_{\lambda x.k(x) \circ \chi} \mid \xi \in \Delta\} \\ &= \sup\{\xi_k \circ \chi \mid \xi \in \Delta\} \\ &= \sup\{\xi_k \mid \xi \in \Delta\} \circ \chi && \text{by continuity of } \circ \\ &= (\sup \Delta)_k \circ \chi\end{aligned}$$
for all $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$, using that composition is Scott-continuous [9, Proposition C.13].

Consequently, $\sup \Delta \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$.

It remains to show that $\sup \Delta$ actually is the supremum of Δ, as our notation has claimed. For any $\xi' \in \Delta$, we have $\xi'_k \leq \sup\{\xi_k \mid \xi \in \Delta\} = (\sup \Delta)_k$, and so $\xi' \leq \sup \Delta$. Instead, assume ξ' is an upper bound for Δ. Then for all k and $\xi \in \Delta$, $\xi'_k \geq \xi_k$ and so $\xi'_k \geq \sup\{\xi_k \mid \xi \in \Delta\} = (\sup \Delta)_k$ and so $\xi' \geq \sup \Delta$. Hence $\sup \Delta$ is the supremum of Δ and so $\mathcal{Q}(\mathscr{A}, \mathscr{B}, X) \in \mathbf{DCPO}$. □

5.1 $\mathcal{Q}$ is a functor

The next two lemmas prove that $\mathcal{Q}$ is a functor $\mathbf{W}^* \times \mathbf{W}^{*\mathrm{op}} \times \mathbf{DCPO} \to \mathbf{DCPO}$. Before proving the equations for functoriality, we must first show that the action on morphisms produces a well-defined function, and that this function is continuous (and hence forms a dcpo morphism).

Lemma 24. *For $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}', \mathscr{B})$, and $f \in \mathbf{DCPO}(X, Y)$, the function $\mathcal{Q}(\phi, \psi, f)$ is well-defined and continuous.*

Proof. To show this function is well-defined, we need that $\mathcal{Q}(\phi, \psi, f)(\xi) \in \mathcal{Q}(\mathscr{A}', \mathscr{B}', Y)$ for $\xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$. We check each condition in turn:

- **Continuity**: Let $K \subseteq \mathbf{DCPO}(Y, \mathbf{W}^*(\mathscr{K}, \mathscr{B}'))$ be directed. Then:

$$\begin{aligned} Q(\phi,\psi,f)(\xi)_{\sup K} &= \phi \circ \xi_{\lambda x.\psi \circ \sup K(f(x))} \\ &= \phi \circ \xi_{\lambda x.\,\sup\{\psi \circ k(f(x)) \mid k \in K\}} && \text{by continuity of } \circ \\ &= \phi \circ \sup\{\xi_{\lambda x.\psi \circ k(f(x))} \mid k \in K\} && \text{by continuity of } \xi \\ &= \sup\{\phi \circ \xi_{\lambda x.\psi \circ k(f(x))} \mid k \in K\} && \text{by continuity of } \circ \end{aligned}$$

  for all k, as required.
- **Subconvexity**: Let $\theta + \rho \leq 1$ and $k, k' \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}'))$. Then we need:

$$\begin{aligned} \phi \circ \xi_{\lambda x.\psi \circ (\theta k(f(x)) + \rho k'(f(x)))} = \quad & \theta(\phi \circ \xi_{\lambda x.\psi \circ k(f(x))}) \\ & + \rho(\phi \circ \xi_{\lambda x.\psi \circ k'(f(x))}) \end{aligned}$$

  but this follows from the linearity of $\circ$ and subconvexity of ξ.
- **Compositionality**: Let $\chi \in \mathbf{W}^*(\mathscr{K}', \mathscr{K})$. Then we need that:

$$\phi \circ \xi_{\lambda x.\psi \circ k(f(x)) \circ \chi} = \phi \circ \xi_{\lambda x.\psi \circ k(f(x))} \circ \chi$$

  but this follows immediately from the compositionality of ξ.

To show continuity, let Δ be a directed set of orchestras and $k \in \mathbf{DCPO}(Y, \mathbf{W}^*(\mathscr{K}, \mathscr{A}'))$. Then:

$$\begin{aligned} Q(\phi,\psi,f)(\sup \Delta)_k &= \phi \circ \sup\{\xi_{\lambda x.\psi \circ k(f(x))} \mid \xi \in \Delta\} \\ &= \sup\{\phi \circ \xi_{\lambda x.\psi \circ k(f(x))} \mid \xi \in \Delta\} && \text{by continuity of } \circ \\ &= (\sup\{Q(\phi,\psi,f)(\xi) \mid \xi \in \Delta\})_k \end{aligned}$$

and so $Q(\phi, \psi, f)$ is continuous. □

Lemma 25. *Q is a functor from $\mathbf{W}^* \times \mathbf{W}^{*\,op} \times \mathbf{DCPO} \to \mathbf{DCPO}$.*

Proof. We first check that Q sends the identity to the identity:

$$Q(\mathrm{id}_{\mathscr{A}}, \mathrm{id}_{\mathscr{B}}, \mathrm{id}_X)(\xi)_k = \mathrm{id}_{\mathscr{A}} \circ \xi_{\lambda x.\,\mathrm{id}_{\mathscr{B}} \circ k(\mathrm{id}_X(x))} = \xi_k$$

Suppose $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{B})$, $\phi' \in \mathbf{W}^*(\mathscr{B}, \mathscr{C})$, $\psi \in \mathbf{W}^*(\mathscr{B}', \mathscr{A}')$, $\psi' \in \mathbf{W}^*(\mathscr{C}', \mathscr{B}')$, $f \in \mathbf{DCPO}(X, Y)$, and $g \in \mathbf{DCPO}(Y, Z)$. Then:

$$Q(\phi',\psi',g)(Q(\phi,\psi,f)(\xi))_k = \phi' \circ \phi \circ \xi_{\lambda x.\psi \circ \psi' \circ k(g(f(x)))} = Q(\phi' \circ \phi, \psi \circ \psi', g \circ f)(\xi)_k$$

for all ξ and k, and so the functor is compatible with composition. □

5.2 The unit is a natural transformation

We next show that the unit η is a natural transformation, whose components are morphisms $X \to Q(\mathscr{A}, \mathscr{A}, X)$ in $\mathbf{DCPO}$

Lemma 26. *The unit η is well-defined, that is for $\mathscr{A} \in \mathbf{W}^*$ and dcpo X that:*

$$\eta_{\mathscr{A}X}(x) \in Q(\mathscr{A}, \mathscr{A}, X)$$

for $x \in X$, and that $\eta_{\mathscr{A}X}$ is continuous.

Proof. As $\eta_{\mathscr{A}X}(x)$ is a Dirac quantum orchestra for each x, $\eta_{\mathscr{A}X}$ is a well-defined function from X to $Q(\mathscr{A}, \mathscr{A}, X)$, so it remains to show it is continuous. Let $D \subseteq X$ be directed, then:

$$\eta_{\mathscr{A}X}(\sup D)_k = k(\sup D) = \sup\{k(x) \mid x \in D\} = \sup\{\eta_{\mathscr{A}X}(x) \mid x \in D\}$$

as required, where the second equality is from the continuity of k. □

Lemma 27. *The unit η is a natural transformation.*

Proof. To show naturality of η, we need naturality in X and dinaturality in $\mathscr{A}$. For the first we require the following to commute:

$$\begin{array}{ccc} X & \xrightarrow{\quad f \quad} & Y \\ {\scriptstyle \eta_{\mathscr{A}X}}\downarrow & & \downarrow{\scriptstyle \eta_{\mathscr{A}Y}} \\ Q(\mathscr{A},\mathscr{A},X) & \xrightarrow{Q(\mathscr{A},\mathscr{A},f)} & Q(\mathscr{A},\mathscr{A},Y) \end{array}$$

for $f \in \mathbf{DCPO}(X,Y)$. We calculate:

$$\eta_{\mathscr{A}Y}(f(x))_k = k(f(x)) = \eta_{\mathscr{A}X}(x)_{\lambda x.k(f(x))} = Q(\mathscr{A},\mathscr{A},f)(\eta_{\mathscr{A}X}(x))_k$$

for all x and k. For $\phi : \mathbf{W}^*(\mathscr{B},\mathscr{A})$, the dinaturality condition is the commutativity of:

$$\begin{array}{ccccc} & & Q(\mathscr{A},\mathscr{A},X) & & \\ & {\scriptstyle \eta_{\mathscr{A}X}}\nearrow & & \searrow{\scriptstyle Q(\mathscr{A},\phi,X)} & \\ X & & & & Q(\mathscr{A},\mathscr{B},X) \\ & {\scriptstyle \eta_{\mathscr{B}X}}\searrow & & \nearrow{\scriptstyle Q(\phi,\mathscr{B},X)} & \\ & & Q(\mathscr{B},\mathscr{B},X) & & \end{array}$$

By again applying both sides to arbitrary x and k and expanding definitions, we get:

$$Q(\mathscr{A},\phi,X)(\eta_{\mathscr{A}X}(x))_k = \phi \circ k(x) = Q(\phi,\mathscr{B},X)(\eta_{\mathscr{B}X}(x))_k$$

Hence, η is natural. □

5.3 The multiplication is a natural transformation

Similarly to the unit, we show that the multiplication is a well-defined natural transformation in two parts, first showing that each component is a well-defined dcpo morphism:

$$Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X)) \to Q(\mathscr{A},\mathscr{C},X)$$

and secondly showing that the required naturality squares hold.

Lemma 28. *The multiplication μ is well-defined: for $\mathscr{A},\mathscr{B},\mathscr{C} \in \mathbf{W}^*$ and dcpo X, we have that:*

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi) \in Q(\mathscr{A},\mathscr{C},X)$$

for $\Xi \in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X))$, and the function $\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}$ is continuous.

Proof. Suppose $\Xi \in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X))$. Then we must show $\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi) \in Q(\mathscr{A},\mathscr{C},X)$. For continuity, if $K \subseteq \mathbf{DCPO}(X,\mathbf{W}^*(\mathscr{K},\mathscr{B}))$ is directed, then:

$$\begin{aligned} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_{\sup K} &= \Xi_{\lambda\xi.\xi_{\sup K}} \\ &= \Xi_{\sup\{\lambda\xi.\xi_k \mid k\in K\}} && \text{by continuity of } \xi \\ &= \sup\{\Xi_{\lambda\xi.\xi_k} \mid k \in K\} && \text{by continuity of } \Xi \\ &= \sup\{\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_k \mid k \in K\} \end{aligned}$$

as required. Subconvexity holds similarly. For compositionality, if $\chi \in \mathbf{W}^*(\mathscr{K}',\mathscr{K})$, then:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_{\lambda x.k(x)\circ\chi} = \Xi_{\lambda\xi.\xi_{\lambda x.k(x)\circ\chi}} = \Xi_{\lambda\xi.\xi_k\circ\chi} = \Xi_{\lambda\xi.\xi_k} \circ \chi = \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_k \circ \chi$$

as required, using the compositionality of each ξ and Ξ. Hence, μ is a well-defined function. In order to show that it is a continuous function we let $\Delta \subseteq Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X))$ be directed. Then:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\sup\Delta)_k = (\sup\Delta)_{\lambda\xi.\xi_k} = \sup\{\Xi_{\lambda\xi.\xi_k} \mid \Xi \in \Delta\} = (\sup\{\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi) \mid \Xi \in \Delta\})_k$$

for any k, completing the proof. □

Lemma 29. *The multiplication μ is a natural transformation.*

Proof. To show μ is natural we need dinaturality in $\mathscr{B}$ and naturality in $\mathscr{A}$, $\mathscr{C}$, and X. We tackle the naturality first, for which we need tho following to commute:

$$\begin{array}{ccc} Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X)) & \xrightarrow{Q(\phi,\mathscr{B},Q(\mathscr{B},\psi,f))} & Q(\mathscr{A}',\mathscr{B},Q(\mathscr{B},\mathscr{C}',Y)) \\ \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}} & & \downarrow{\scriptstyle \mu_{\mathscr{A}',\mathscr{B},\mathscr{C}',Y}} \\ Q(\mathscr{A},\mathscr{C},X) & \xrightarrow{Q(\phi,\psi,f)} & Q(\mathscr{A}',\mathscr{C}',Y) \end{array}$$

for $\phi : \mathbf{W}^*(\mathscr{A},\mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{C}',\mathscr{C})$, and $f : X \to Y$. For each Ξ and k, we have:

$$\mu_{\mathscr{A}',\mathscr{B},\mathscr{C}',Y}(Q(\phi,\mathscr{B},Q(\mathscr{B},\psi,f))(\Xi))_k = \phi \circ \Xi_{\lambda\xi.\xi_{\lambda x.\psi\circ k(x)}} = Q(\phi,\psi,f)(\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi))_k$$

simply by expanding definitions on each side. For dinaturality, let $\phi \in \mathbf{W}^*(\mathscr{B}',\mathscr{B})$. Then the following diagram must commute:

$$\begin{array}{ccccc} & & Q(\mathscr{A},\mathscr{B}',Q(\mathscr{B}',\mathscr{C},X)) & & \\ & {\scriptstyle Q(\mathscr{A},\phi,Q(\mathscr{B}',\mathscr{C},X))}\nearrow & & \searrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B}',\mathscr{C},X}} & \\ Q(\mathscr{A},\mathscr{B},Q(\mathscr{B}',\mathscr{C},X)) & & & & Q(\mathscr{A},\mathscr{C},X) \\ & {\scriptstyle Q(\mathscr{A},\mathscr{B},Q(\phi,\mathscr{C},X))}\searrow & & \nearrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}} & \\ & & Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X)) & & \end{array}$$

Then again by expanding definitions we get:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(Q(\mathscr{A},\mathscr{B},Q(\phi,\mathscr{C},X))(\Xi))_k = \Xi_{\lambda\xi.\phi\circ\xi_k} = \mu_{\mathscr{A},\mathscr{B}',\mathscr{C},X}(Q(\mathscr{A},\phi,Q(\mathscr{B}',\mathscr{C},X))(\Xi))_k$$

for each Ξ and k and so μ is natural as required. □

5.4 Q is a monad

We can now show that Q does indeed form a monad.

Theorem 30. *(Q,η,μ) is a $\mathbf{W}^{*^{op}}$-parameterised monad on* **DCPO**.

Proof. We check each monad law in turn. The first unitality law states that:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{B},X} \circ Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X}) = \mathrm{id}_{Q(\mathscr{A},\mathscr{B},X)}$$

Taking $\xi \in Q(\mathscr{A},\mathscr{B},X)$ and $k \in \mathbf{DCPO}(X,\mathbf{W}^*(\mathscr{K},\mathscr{B}))$:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{B},X}(Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X})(\xi))_k = Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X})(\xi))_{\lambda\xi'.\xi'_k} = \xi_{\lambda x.\eta_{\mathscr{A}X}(x)_k} = \xi_k$$

The second unitality law is $\mu_{\mathscr{A},\mathscr{A},\mathscr{B},X} \circ \eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B},X)} = \mathrm{id}_{Q(\mathscr{A},\mathscr{B},X)}$. Taking the same ξ and k we have:

$$\mu_{\mathscr{A},\mathscr{A},\mathscr{B},X}(\eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B},X)}(\xi))_k = \eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B}X)}(\xi)_{\lambda\xi'.\xi'_k} = \xi_k$$

The associativity monad law is the commutativity of the following diagram:

$$\begin{array}{ccc} Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X))) & \xrightarrow{\mu_{\mathscr{A},\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)}} & Q(\mathscr{A},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)) \\ \downarrow{\scriptstyle Q(\mathscr{A},\mathscr{B},\mu_{\mathscr{B},\mathscr{C},\mathscr{D},X})} & & \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{C},\mathscr{D},X}} \\ Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{D},X)) & \xrightarrow{\mu_{\mathscr{A},\mathscr{B},\mathscr{D},X}} & Q(\mathscr{A},\mathscr{D},X) \end{array}$$

By expanding definitions on both sides we have:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{D},X}(Q(\mathscr{A},\mathscr{B},\mu_{\mathscr{B},\mathscr{C},\mathscr{D},X})(\aleph)) = \aleph_{\lambda\Xi.\Xi_{\lambda\xi.\xi_k}} = \mu_{\mathscr{A},\mathscr{C},\mathscr{D},X}(\mu_{\mathscr{A},\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)}(\aleph))_k$$

for $\aleph \in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)))$ and $k \in \mathbf{DCPO}(X,\mathbf{W}^*(\mathscr{K},\mathscr{D}))$, and so the square commutes. With these three laws proven, we have shown that Q is a monad. □

5.5 Strength of the monad

Finally, we show that the monad we have constructed is strong. Much like Sections 5.2 and 5.3, we must first show that τ is well-defined before showing that it is a natural transformation.

Lemma 31. *The strength τ is well-defined, that is:*

$$\tau_{X,\mathscr{A},\mathscr{B},Y} \in \mathbf{DCPO}(X \times \mathcal{Q}(\mathscr{A},\mathscr{B},Y), \mathcal{Q}(\mathscr{A},\mathscr{B},X\times Y))$$

for dcpos X and Y, and $\mathscr{A},\mathscr{B} \in \mathbf{W}^*$.

Proof. We first show that $\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi) \in \mathcal{Q}(\mathscr{A},\mathscr{B},X\times Y)$ for $x \in X$ and $\xi \in \mathcal{Q}(\mathscr{A},\mathscr{B},Y)$. Let $K \subseteq \mathbf{DCPO}(X\times Y, \mathbf{W}^*(\mathscr{K},\mathscr{B}))$ be directed. Then:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}(\xi)_{\sup K} = \xi_{\lambda y.\,\sup K(x,y)} = \sup\{\xi_{\lambda y.k(x,y)} \mid k \in K\} = \sup\{\tau_{X,\mathscr{A},\mathscr{B},Y}(\xi)_k \mid k \in K\}$$

Subconvexity holds similarly: given θ,ρ such that $\theta+\rho \le 1$ and k,k', we have:

$$\begin{aligned}\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_{\lambda z.\theta k(z)+\rho k'(z)} &= \xi_{\lambda y.\theta k(x,y)+\rho k'(x,y)}\\ &= \theta\xi_{\lambda y.k(x,y)} + \rho\xi_{\lambda y.k'(x,y)} && \text{by subconvexity of } \xi\\ &= \lambda\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_k + \rho\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_{k'}\end{aligned}$$

For compositionality, we have:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_{\lambda z.k(z)\circ\chi} = \xi_{\lambda y.k(x,y)\circ\chi} = \xi_{\lambda y.k(x,y)} \circ \chi = \tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_k \circ \chi$$

for $\chi \in \mathbf{W}^*(\mathscr{K}',\mathscr{K})$ and $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K},\mathscr{B}))$, using the compositionality of ξ. Therefore, $\tau_{X,\mathscr{A},\mathscr{B},Y}(\xi) \in \mathcal{Q}(\mathscr{A},\mathscr{B},X\times Y)$ as required.

To show $\tau_{X,\mathscr{A},\mathscr{B},Y} \in \mathbf{DCPO}(X\times\mathcal{Q}(\mathscr{A},\mathscr{B},Y), \mathcal{Q}(\mathscr{A},\mathscr{B},X\times Y))$, it remains to show that it is continuous, for which it is sufficient to show continuity in each argument separately. First let $D \subseteq X$ be directed. Then for fixed ξ and k:

$$\begin{aligned}\tau_{X,\mathscr{A},\mathscr{B},Y}(\sup D,\xi)_k &= \xi_{\lambda y.k(\sup D,y)}\\ &= \xi_{\sup\{\lambda y.k(d,y)\mid d\in D\}} && \text{by continuity of } k\\ &= \sup\{\xi_{\lambda y.k(d,y)} \mid d \in D\} && \text{by continuity of } \xi\\ &= \sup\{\tau_{X,\mathscr{A},\mathscr{B},Y}(d,\xi) \mid d\in D\}_k\end{aligned}$$

Now let $\Delta \subseteq \mathcal{Q}(\mathscr{A},\mathscr{B},Y)$ be directed. Then for fixed x and k:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\sup\Delta)_k = (\sup\Delta)_{\lambda y.k(x,y)} = \{\xi_{\lambda y.k(x,y)} \mid \xi\in\Delta\} = \sup\{\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi) \mid \xi\in\Delta\}_k$$

by the definition of the supremum on $\mathcal{Q}(\mathscr{A},\mathscr{B},Y)$. Hence, τ is well-defined. □

Lemma 32. *τ is a natural transformation.*

Proof. Let $f : X \to X'$, $g : Y \to Y'$, $\phi : \mathbf{W}^*(\mathscr{A},\mathscr{A}')$, and $\psi : \mathbf{W}^*(\mathscr{B}',\mathscr{B})$. Then the following must commute:

$$\begin{array}{ccc} X\times\mathcal{Q}(\mathscr{A},\mathscr{B},Y) & \xrightarrow{f\times\mathcal{Q}(\phi,\psi,g)} & X'\times\mathcal{Q}(\mathscr{A}',\mathscr{B}',Y') \\ \downarrow{\scriptstyle \tau_{X,\mathscr{A},\mathscr{B},Y}} & & \downarrow{\scriptstyle \tau_{X',\mathscr{A}',\mathscr{B}',Y'}} \\ \mathcal{Q}(\mathscr{A},\mathscr{B},X\times Y) & \xrightarrow{\mathcal{Q}(\phi,\psi,f\times g)} & \mathcal{Q}(\mathscr{A}',\mathscr{B}',X'\times Y') \end{array}$$

For $x\in X$, $\xi\in\mathcal{Q}(\mathscr{A},\mathscr{B},Y)$, and $k\in\mathbf{DCPO}(X'\times Y', \mathbf{W}^*(\mathscr{K},\mathscr{B}'))$, we get:

$$\begin{aligned}\tau_{X',\mathscr{A}',\mathscr{B}',Y'}((f\times\mathcal{Q}(\phi,\psi,g))(x,\xi))_k &= \phi\circ\xi_{\lambda y.\psi\otimes\mathscr{E}\circ k(f(x),g(y))}\\ &= \mathcal{Q}(\phi,\psi,f\times g)(\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi))_k\end{aligned}$$

by expanding definitions and so τ is a natural transformation. □

Theorem 33. *$(\mathcal{Q}, \eta, \mu, \tau)$ is a $\mathbf{W}^{*^{op}}$-parameterised strong monad on* **DCPO**.

Proof. From Theorem 30, we have that $(\mathcal{Q}, \eta, \mu)$ is a monad. It remains to prove the appropriate monad laws for the strength. Letting $1 = \{*\}$ and $L_X : 1 \times X \to X$ be the left unitor isomorphism, we need:

$$\mathcal{Q}(\mathscr{A}, \mathscr{B}, L_X) \circ \tau_{1,\mathscr{A},\mathscr{B},X} = L_{\mathcal{Q}(\mathscr{A},\mathscr{B},X)}$$

For $\xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ and $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$, we have:

$$\mathcal{Q}(\mathscr{A}, \mathscr{B}, L_X)(\tau_{1,\mathscr{A},\mathscr{B},X}(*, \xi))_k = \tau_{1,\mathscr{A},\mathscr{B},X}(*, \xi)_{\lambda z.k(L_X(z))} = \xi_{\lambda x.k(L_X(*,x))} = \xi_k = L_{\mathcal{Q}(\mathscr{A},\mathscr{B},X)}(*, \xi_k)$$

For compatibility with the associator $\alpha_{X,Y,Z} : (X \times Y) \times Z \to X \times (Y \times Z)$, the following must commute:

$$\begin{array}{ccccc}
(X \times Y) \times \mathcal{Q}(\mathscr{A}, \mathscr{B}, Z) & & \xrightarrow{\tau_{X\times Y,\mathscr{A},\mathscr{B},Z}} & & \mathcal{Q}(\mathscr{A}, \mathscr{B}, (X \times Y) \times Z) \\
\downarrow{\scriptstyle \alpha_{X,Y,\mathcal{Q}(\mathscr{A},\mathscr{B},Z)}} & & & & \downarrow{\scriptstyle \mathcal{Q}(\mathscr{A},\mathscr{B},\alpha_{X,Y,Z})} \\
X \times (Y \times \mathcal{Q}(\mathscr{A}, \mathscr{B}, Z)) & \xrightarrow{\mathrm{id}_X \times \tau_{Y,\mathscr{A},\mathscr{B},Z}} & X \times \mathcal{Q}(\mathscr{A}, \mathscr{B}, Y \times Z) & \xrightarrow{\tau_{X,\mathscr{A},\mathscr{B},Y\times Z}} & \mathcal{Q}(\mathscr{A}, \mathscr{B}, X \times (Y \times Z))
\end{array}$$

For $x \in X$, $y \in Y$, $\xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, Z)$, and $k \in \mathbf{DCPO}(X \times (Y \times Z), \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$, we have:

$$\begin{aligned}
\tau_{X,\mathscr{A},\mathscr{B},Y\times Z}((\mathrm{id}_X \times \tau_{Y,\mathscr{A},\mathscr{B},Z})(\alpha_{X,Y,\mathcal{Q}(\mathscr{A},\mathscr{B},Z)}((x, y), \xi)))_k &= \xi_{\lambda z.k(x,(y,z))} \\
&= \mathcal{Q}(\mathscr{A}, \mathscr{B}, \alpha_{X,Y,Z})(\tau_{X\times Y,\mathscr{A},\mathscr{B},Z}((x, y), \xi))_k
\end{aligned}$$

by unfolding definitions. For compatibility with the unit we need:

$$\tau_{X,\mathscr{A},\mathscr{A},Y} \circ (\mathrm{id}_X \times \eta_{\mathscr{A}Y}) = \eta_{\mathscr{A}X\times Y}$$

Given $x \in X$, $y \in Y$, and $k \in \mathbf{DCPO}(X \times Y, \mathbf{W}^*(\mathscr{K}, \mathscr{A}))$:

$$\tau_{X,\mathscr{A},\mathscr{A},Y}(x, \eta_{\mathscr{A}Y}(y))_k = \eta_{\mathscr{A}Y}(y)_{\lambda y.k(x,y)} = k(x, y) = \eta_{\mathscr{A}X\times Y}(x, y)_k$$

Finally for compatibility with μ we need the following to commute:

$$\begin{array}{ccccc}
X \times \mathcal{Q}(\mathscr{A}, \mathscr{B}, \mathcal{Q}(\mathscr{B}, \mathscr{C}, Y)) & \xrightarrow{\tau_{X,\mathscr{A},\mathscr{B},\mathcal{Q}(\mathscr{B},\mathscr{C},Y)}} & \mathcal{Q}(\mathscr{A}, \mathscr{B}, X \times \mathcal{Q}(\mathscr{B}, \mathscr{C}, Y)) & \xrightarrow{\mathcal{Q}(\mathscr{A},\mathscr{B},\tau_{X,\mathscr{B},\mathscr{C},Y})} & \mathcal{Q}(\mathscr{A}, \mathscr{B}, \mathcal{Q}(\mathscr{B}, \mathscr{C}, X \times Y)) \\
\downarrow{\scriptstyle \mathrm{id}_X \times \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}} & & & & \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X\times Y}} \\
X \times \mathcal{Q}(\mathscr{A}, \mathscr{C}, Y) & & \xrightarrow{\tau_{X,\mathscr{A},\mathscr{C},Y}} & & \mathcal{Q}(\mathscr{A}, \mathscr{C}, X \times Y)
\end{array}$$

So taking $x \in X$, $\Xi \in \mathcal{Q}(\mathscr{A}, \mathscr{B}, \mathcal{Q}(\mathscr{B}, \mathscr{C}, Y))$, and $k \in \mathbf{DCPO}(X \times Y, \mathbf{W}^*(\mathscr{K}, \mathscr{C}))$:

$$\begin{aligned}
\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X\times Y}(\mathcal{Q}(\mathscr{A}, \mathscr{B}, \tau_{X,\mathscr{B},\mathscr{C},Y})(\tau_{X,\mathscr{A},\mathscr{B},\mathcal{Q}(\mathscr{B},\mathscr{C},Y)}(x, \Xi)))_k &= \Xi_{\lambda\xi.\xi_{\lambda y.k(x,y)}} \\
&= \tau_{X,\mathscr{A},\mathscr{C},Y}(x, \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\Xi))_k
\end{aligned}$$

again by simply unfolding definitions on both sides. Hence, all of the monad laws hold and $(\mathcal{Q}, \eta, \mu, \tau)$ is a $\mathbf{W}^{*^{\mathrm{op}}}$-parameterised strong monad. □

6 Discussion and future work

In this paper, we presented the *quantum orchestra* monad, which captures quantum side effects by describing them as the most general form of physical evolution of quantum systems, subunital completely positive maps on W*-algebras. As the monad acts on the category **DCPO** of directed complete partially ordered sets, it naturally enables the interpretation of hybrid classical-quantum programs, including those relying on unbounded recursion.

The quantum orchestra monad heavily draws inspiration from *continuation passing-style* monads, in agreement with other monad constructions based on the quantum instrument formalism [6, 15]. This seems to indicate that quantum effects of programs are most naturally described in CPS,

rather than in direct style. We leave the question whether a quantum orchestra-like monad with direct-style flavour is possible for future investigation.

Order structure of quantum orchestras and $\mathbf{W}^*(\mathscr{B},\mathscr{A})$*-valued valuations.* As hinted at in Section 4, one main advantage of defining the quantum orchestra monad in CPS is the possibility of defining the order of quantum orchestras pointwise for all continuations. Valuations that take values in some $\mathbf{W}^*(\mathscr{B},\mathscr{A})$ can be seen as a direct-style alternative to quantum orchestras, and in fact orchestras restrict to valuations, by considering the components associated to indicator functions. It remains unclear how the pointwise order of these valuations however relates to the order of quantum orchestras, as the former only "tests" on indicator functions as opposed to all continuations, and thus has the potential to be insufficiently coarse to render integration continuous. Interestingly, it is known that these two orders coincide in the case of scalar-valued valuations, a consequence of the max-flow min-cut theorem [24]. While the additional lattice-structure of commutative W*-algebras (their self-adjoint parts form in fact Dedekind complete Riesz spaces [14]) makes it believable that an analogous correspondence might hold more generally for valuations and quantum orchestras over commutative W*-algebras, thus allowing for continuous integration along valuations, we leave this problem, as well as the more general non-commutative case for future research.

Relation to the probabilistic powerdomain monad. Given any $X \in \mathbf{DCPO}$, we observe that all orchestras $\xi \in \mathcal{Q}(\mathbb{C},\mathbb{C},X)$ restrict to $[0,1]$-valued valuations on indicator functions. Compositionality of quantum orchestras guarantees that this mapping is injective, and the pointwise nature of the order of quantum orchestras implies that this mapping is continuous. The structure of such quantum orchestras is thus entirely determined by the underlying valuations. Vice versa, any $[0,1]$-valued valuation on X yields a unique continuous integral for continuous scalar-valued functions on X which indeed gives rise a full, consistent orchestra. Perhaps surprisingly, keeping in mind that quantum orchestras were entirely defined in CPS, it turns out that the thus found relation between the simplest quantum orchestras and probabilistic powerdomains is in fact an *isomorphism of monads*. The full proof of this claim can be found in Appendix D.

Theorem 34. $\mathcal{Q}(\mathbb{C},\mathbb{C},-)$ *and the probabilistic powerdomain monad* $\mathcal{V}$ *are equivalent.*

Note that the commutativity of the probabilistic powerdomain monad, which is equivalent to a Fubini property of the valuations integral, has been an open problem since its discovery [7], and thus this question also affects $\mathcal{Q}(\mathbb{C},\mathbb{C},-)$. Beyond this special case, $\mathcal{Q}(\mathscr{A},\mathscr{A},-)$ is not commutative for any W*-algebra $\mathscr{A}$ which is not equal to $\mathbb{C}$, which reflects the fact that the composition of morphisms in $\mathbf{W}^*(\mathscr{A},\mathscr{A})$ is non-commutative, even if $\mathscr{A}$ is abelian. This perspective indicates that the multiplication of the quantum orchestra monad implements a form of *non-commutative integral*.

Denotational semantics for existing languages. In this work, we have seen how the new quantum orchestra monad could be used to give semantics for a prototype hybrid classical-quantum language, exhibiting the monad's versatility. This makes it a natural and flexible tool to further develop the semantics of existing languages in the QRAM model. In this context, it would be interesting to explore the possibility of accompanying orchestra-based denotational semantics with a compatible operational semantics. We leave this question for future work.

Acknowledgments

AR was funded by UKRI grant EP/X025551/1: Rubber DUQ: Flexible Dynamic Universal Quantum programming. DL acknowledges support from the Quantum Advantage Pathfinder research program within the UK's National Quantum Computing Center. RIB was supported by the Engineering

and Physical Sciences Research Council grant reference EP/Z002230/1: (De)constructing quantum software (DeQS).

References

[1] Paetznick Adam and Krysta M. Svore. 2014. Repeat-Until-Success: Non-deterministic Decomposition of Single-Qubit Unitaries. *Quantum Information and Computation* 14, 15&16 (Nov. 2014), 1277–1301. doi:10.26421/qic14.15-16-2

[2] Thorsten Altenkirch and Alexander S Green. 2010. The quantum IO monad. *Semantic Techniques in Quantum Computation* (2010), 173–205. doi:10.1017/CBO9781139193313

[3] Robert Atkey. 2009. Parameterised Notions of Computation. *Journal of Functional Programming* 19, 3-4 (July 2009), 335–376. doi:10.1017/S095679680900728X

[4] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R. Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, Robert A. Lang, Christina Lee, Thomas Loke, Angus Lowe, Keri McKiernan, Johannes Jakob Meyer, J. A. Montañez-Barrera, Romain Moyard, Zeyue Niu, Lee James O'Riordan, Steven Oud, Ashish Panigrahi, Chae-Yeun Park, Daniel Polatajko, Nicolás Quesada, Chase Roberts, Nahum Sá, Isidor Schoch, Borun Shi, Shuli Shu, Sukin Sim, Arshpreet Singh, Ingrid Strandberg, Jay Soni, Antal Száva, Slimane Thabet, Rodrigo A. Vargas-Hernández, Trevor Vincent, Nicola Vitucci, Maurice Weber, David Wierichs, Roeland Wiersema, Moritz Willmann, Vincent Wong, Shaoming Zhang, and Nathan Killoran. 2022. PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations. arXiv:1811.04968 [physics, physics:quant-ph] doi:10.48550/arXiv.1811.04968

[5] Benjamin Bichsel, Maximilian Baader, Timon Gehr, and Martin Vechev. 2020. Silq: A High-Level Quantum Language with Safe Uncomputation and Intuitive Semantics. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2020)*. Association for Computing Machinery, New York, NY, USA, 286–300. doi:10.1145/3385412.3386007

[6] Robert I. Booth, Dominik Leichtle, Alex Rice, and Kim Worrall. 2026. Composing Quantum Instruments. arXiv:2606.28291 [quant-ph] doi:10.48550/arXiv.2606.28291

[7] Titouan Carette, Louis Lemonnier, and Vladimir Zamdzhiev. 2023. Central Submonads and Notions of Computation: Soundness, Completeness and Internal Languages. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*. IEEE, 1–13. doi:10.1109/LICS56636.2023.10175687

[8] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. 2021. Variational Quantum Algorithms. *Nature Reviews Physics* 3, 9 (Sept. 2021), 625–644. doi:10.1038/s42254-021-00348-9

[9] Kenta Cho. 2014. Semantics for a Quantum Programming Language by Operator Algebras. *New Generation Computing* 34 (2014), 25–68. doi:10.1007/s00354-016-0204-3

[10] Andrew Cross, Ali Javadi-Abhari, Thomas Alexander, Niel De Beaudrap, Lev S. Bishop, Steven Heidel, Colm A. Ryan, Prasahnt Sivarajah, John Smolin, Jay M. Gambetta, and Blake R. Johnson. 2022. OpenQASM 3: A Broader and Deeper Quantum Assembly Language. *ACM Transactions on Quantum Computing* 3, 3 (Sept. 2022), 12:1–12:50. doi:10.1145/3505636

[11] E Brian Davies and John T Lewis. 1970. An operational approach to quantum probability. *Communications in Mathematical Physics* 17, 3 (1970), 239–260. doi:10.1007/BF01647093

[12] David Elieser Deutsch. 1989. Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425, 1868 (09 1989), 73–90. arXiv:https://royalsocietypublishing.org/rspa/article-pdf/425/1868/73/67392/rspa.1989.0099.pdf doi:10.1098/rspa.1989.0099

[13] Dhruv Devulapalli, Eddie Schoute, Aniruddha Bapat, Andrew M. Childs, and Alexey V. Gorshkov. 2024. Quantum Routing with Teleportation. *Physical Review Research* 6, 3 (Sept. 2024), 033313. doi:10.1103/PhysRevResearch.6.033313

[14] Peter G. Dodds. 1974. The order dual of an abelian von Neumann algebra. *Journal of the Australian Mathematical Society* 18, 2 (1974), 153–160. doi:10.1017/S1446788700019868

[15] Tobias Fritz. 2026. The quantum instrument monad. arXiv:2606.27805 [cs.LO] doi:10.48550/arXiv.2606.27805

[16] Marco Gaboardi, Shin-ya Katsumata, Dominic Orchard, Flavien Breuvart, and Tarmo Uustalu. 2016. Combining effects and coeffects via grading. In *Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming* (Nara Japan, 2016-09-04). ACM, 476–489. doi:10.1145/2951913.2951939

[17] Michèle Giry. 1982. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, B. Banaschewski (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 68–85. doi:10.1007/BFb0092872

[18] Cassandra Granade and Nathan Wiebe. 2022. Using Random Walks for Iterative Phase Estimation. arXiv:2208.04526 [quant-ph] doi:10.48550/arXiv.2208.04526

[19] Alexander S. Green, Peter LeFanu Lumsdaine, Neil J. Ross, Peter Selinger, and Benoît Valiron. 2013. Quipper: a scalable quantum programming language. *ACM SIGPLAN Notices* 48, 6 (June 2013), 333–342. doi:10.1145/2499370.2462177

[20] Chris Heunen and Robin Kaarsgaard. 2022. Quantum information effects. *Proc. ACM Program. Lang.* 6, POPL, Article 2 (Jan. 2022), 27 pages. doi:10.1145/3498663

[21] Mathieu Huot and Sam Staton. 2019. Quantum channels as a categorical completion. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019*. IEEE, 1–13. doi:10.1109/LICS.2019.8785700

[22] David Ittah, Thomas Häner, Vadym Kliuchnikov, and Torsten Hoefler. 2022. QIRO: A Static Single Assignment-based Quantum Program Representation for Optimization. *ACM Transactions on Quantum Computing* 3, 3 (Sept. 2022), 1–32. doi:10.1145/3491247

[23] Xiaodong Jia, Andre Kornell, Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhiev. 2022. Semantics for variational Quantum programming. *Proc. ACM Program. Lang.* 6, POPL (2022), 1–31. doi:10.1145/3498687

[24] C. Jones and G. Plotkin. 1989. A probabilistic powerdomain of evaluations. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science* (Pacific Grove, California, USA). IEEE Press, 186–195.

[25] Ohad Kammar, Jack Liell-Cock, Sam Lindley, Cristina Matache, and Sam Staton. 2026. An Equational Axiomatization of Dynamic Threads via Algebraic Effects: Presheaves on Finite Relations, Labelled Posets, and Parameterized Algebraic Theories. *Proc. ACM Program. Lang.* 10, POPL, Article 64 (Jan. 2026), 29 pages. doi:10.1145/3776706

[26] Shin-ya Katsumata. 2014. Parametric effect monads and semantics of effect systems. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (San Diego California USA, 2014-01-08). ACM, 633–645. doi:10.1145/2535838.2535846

[27] Shin-ya Katsumata, Dylan McDermott, Tarmo Uustalu, and Nicolas Wu. 2022. Flexible presentations of graded monads. 6 (2022), 902–930. Issue ICFP. doi:10.1145/3547654

[28] Klaus Keimel and Gordon D. Plotkin. 2017. Mixed powerdomains for probability and nondeterminism. Volume 13, Issue 1 (2017). doi:10.23638/LMCS-13(1:2)2017

[29] E. Knill. 1996. *Conventions for Quantum Pseudocode.* Technical Report LA-UR-96-2724. Los Alamos National Laboratory. doi:10.2172/366453

[30] Mark Koch, Agustín Borgna, Seyon Sivarajah, Alan Lawrence, Alec Edgington, Douglas Wilson, Ross Duncan, Craig Roy, Luca Mondada, and Lukas Heidemann. 2025. HUGR: A Quantum-Classical Intermediate Representation. *PlanQC 2025* (2025). doi:10.48550/arXiv.2510.11420

[31] Mark Koch, Alan Lawrence, Kartik Singhal, Seyon Sivarajah, and Ross Duncan. 2025. GUPPY: Pythonic Quantum-Classical Programming. arXiv:2510.12582 [cs.PL] doi:10.48550/arXiv.2510.12582

[32] Paul Blain Levy. 2003. *Call-By-Push-Value.* Springer Netherlands, Dordrecht. doi:10.1007/978-94-007-0954-6

[33] Paul Blain Levy, John Power, and Hayo Thielecke. 2003. Modelling environments in call-by-value programming languages. *Information and Computation* 185, 2 (2003), 182–210. doi:10.1016/S0890-5401(03)00088-9

[34] Eugenio Moggi. 1991. Notions of computation and monads. *Inf. Comput.* 93, 1 (July 1991), 55–92. doi:10.1016/0890-5401(91)90052-4

[35] Paolo Perrone. 2024. *Starting Category Theory.* WORLD SCIENTIFIC. doi:10.1142/13670

[36] Christophe Piveteau and David Sutter. 2024. Circuit Knitting With Classical Communication. *IEEE Trans. Inf. Theor.* 70, 4 (April 2024), 2734–2745. doi:10.1109/TIT.2023.3310797

[37] QIR Alliance. 2021. *QIR Specification.* Retrieved 2026-07-09 from https://github.com/qir-alliance/qir-spec/tree/main/specification

[38] Robert Raussendorf and Hans J. Briegel. 2001. A One-Way Quantum Computer. *Physical Review Letters* 86, 22 (May 2001), 5188–5191. doi:10.1103/PhysRevLett.86.5188

[39] Francisco Rios and Peter Selinger. 2018. A Categorical Model for a Quantum Circuit Description Language (Extended Abstract). *Electronic Proceedings in Theoretical Computer Science* 266 (Feb. 2018), 164–178. doi:10.4204/eptcs.266.11

[40] Joschka Roffe. 2019. Quantum error correction: an introductory guide. *Contemporary Physics* 60, 3 (2019), 226–245. doi:10.1080/00107514.2019.1667078

[41] Shôichirô Sakai and Shãoichirão Sakai. 1971. *C^*-algebras and W^*-algebras.* Classics in Mathematics, Vol. 60. Springer.

[42] Ken Sakayori, Andrea Colledan, and Ugo Dal Lago. 2026. On Circuit Description Languages, Indexed Monads, and Resource Analysis. *Proc. ACM Program. Lang.* 10, POPL, Article 24 (Jan. 2026), 30 pages. doi:10.1145/3776666

[43] Peter Selinger. 2007. Dagger Compact Closed Categories and Completely Positive Maps: (Extended Abstract). *Electronic Notes in Theoretical Computer Science* 170 (2007), 139–163. doi:10.1016/j.entcs.2006.12.018 Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL 2005).

[44] Peter Selinger and Benoît Valiron. 2005. *A Lambda Calculus for Quantum Computation with Classical Control.* Springer Berlin Heidelberg, 354–368. doi:10.1007/11417170_26

[45] Sam Staton. 2015. Algebraic Effects, Linearity, and Quantum Programming Languages. *SIGPLAN Not.* 50, 1 (Jan. 2015), 395–406. doi:10.1145/2775051.2676999

[46] Viggo Stoltenberg-Hansen, Ingrid Lindström, and Edward R Griffor. 1994. *Mathematical theory of domains.* Number 22. Cambridge University Press.
[47] Alfred Tarski. 1955. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* 5, 2 (1955), 285–309. doi:10.2140/pjm.1955.5.285
[48] Joel J. Wallman and Joseph Emerson. 2016. Noise Tailoring for Scalable Quantum Computation via Randomized Compiling. *Physical Review A* 94, 5 (Nov. 2016), 052325. doi:10.1103/PhysRevA.94.052325

A Toy Language - Full Description

The final typing rules are given below. We first begin with the rules for values.

$$\frac{x : X \in \Gamma}{\Gamma \vdash^v x : X} \qquad \frac{}{\Gamma \vdash^v () : 1} \qquad \frac{\Gamma \vdash^v v_1 : X \qquad \Gamma \vdash^v v_2 : Y}{\Gamma \vdash^v (v_1, v_2) : X \times Y} \qquad \frac{\Gamma \vdash^v v : X \times Y}{\Gamma \vdash^v \mathsf{fst}\, v : X} \qquad \frac{\Gamma \vdash^v v : X \times Y}{\Gamma \vdash^v \mathsf{snd}\, v : Y}$$

$$\frac{}{\Gamma \vdash^v \mathsf{True} : \mathsf{bool}} \qquad \frac{}{\Gamma \vdash^v \mathsf{False} : \mathsf{bool}} \qquad \frac{\Gamma, x : X \vdash^e_n e : Y}{\Gamma \vdash^v \lambda x.e : X \to_n Y} \qquad \frac{\mathrm{G} \in \{\mathrm{H}, \mathrm{S}, \mathrm{X}, \mathrm{T}\}}{\Gamma \vdash^v \mathrm{G} : \mathsf{qid} \to_0 1}$$

$$\frac{}{\Gamma \vdash^v \mathrm{CX} : \mathsf{qid} \times \mathsf{qid} \to_0 1} \qquad \frac{}{\Gamma \vdash^v \textsc{meas} : \mathsf{qid} \to_0 \mathsf{bool}} \qquad \frac{}{\Gamma \vdash^v \textsc{alloc} : 1 \to \mathsf{qid}}$$

And then we have the rules for expressions.

$$\frac{\Gamma \vdash^v v : X}{\Gamma \vdash^e_0 \mathsf{return}\, v : X} \qquad \frac{\Gamma \vdash^e_{n_1} e_1 : X \qquad \Gamma, x : X \vdash^e_{n_2} e_2 : Y}{\Gamma \vdash^e_{n_1+n_2} \mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2 : Y} \qquad \frac{\Gamma \vdash^e_{n_1} e_1 : X \qquad \Gamma \vdash^e_{n_2} e_2 : Y}{\Gamma \vdash^e_{n_1+n_2} e_1; e_2 : Y}$$

$$\frac{\Gamma \vdash^v v_1 : X \to_n Y \qquad \Gamma \vdash^v v_2 : X}{\Gamma \vdash^e_n v_1(v_2) : Y} \qquad \frac{\Gamma \vdash^v v : \mathsf{bool} \qquad \Gamma \vdash^e_n e_1 : X \qquad \Gamma \vdash^e_n e_2 : X}{\Gamma \vdash^e_n \mathsf{if}\, v \,\mathsf{then}\, e_1 \,\mathsf{else}\, e_2 : X}$$

$$\frac{\Gamma \vdash^v f : (1 \to_0 X) \to_0 X}{\Gamma \vdash^e_0 \mathsf{fix}\, f : X}$$

The semantics for types are parameterised over the number of qubits available and are given by:

$$[\![1]\!]_m = \{()\} \qquad [\![X \times Y]\!]_m = [\![X]\!]_m \times [\![Y]\!]_m \qquad [\![\mathsf{bool}]\!]_m = \{\mathsf{True}, \mathsf{False}\}$$

$$[\![X \to_n Y]\!]_m = \forall m' \geq m.\mathbf{DCPO}([\![X]\!]_{m'}, \mathcal{Q}(\mathcal{B}(\mathcal{H}^{m'}), \mathcal{B}(\mathcal{H}^{m'+n}), [\![Y]\!]_{m'+n}))$$

$$[\![\mathsf{qid}]\!]_m = \{0, \ldots, m-1\}$$

where we let $\mathcal{H} = \mathbb{C}^2$ and the semantics of contexts is given by:

$$[\![x_1 : X_1, \ldots, x_n : X_n]\!]_m = [\![X_1]\!]_m \times \cdots \times [\![X_n]\!]_m$$

We note that the definition of the function type is quantified over the number of qubits. This trick allows us to define a function:

$$\iota_{X,m,m'} : [\![X]\!]_m \to [\![X]\!]_{m'}$$

which can be extended from types X to contexts Γ.

The semantics for a value $\Gamma \vdash^v v : X$ is a (continuous) function:

$$v_m \in \mathbf{DCPO}([\![\Gamma]\!]_m, [\![X]\!]_m)$$

We define it for each constructor as follows:

$$\begin{aligned}
[\![x]\!]_m(\rho) &= \rho_x & [\![(v_1, v_2)]\!]_m(\rho) &= ([\![v_1]\!]_m(\rho), [\![v_2]\!]_m(\rho)) \\
[\![()]\!]_m(\rho) &= () & [\![\mathsf{fst}\, v]\!]_m(\rho) &= \pi_1([\![v]\!]_m(\rho)) \\
[\![\mathsf{True}]\!]_m(\rho) &= \mathsf{True} & [\![\mathsf{snd}\, v]\!]_m(\rho) &= \pi_2([\![v]\!]_m(\rho)) \\
[\![\mathsf{False}]\!]_m(\rho) &= \mathsf{False} & [\![\lambda x.e]\!]_m(\rho)(\rho_x) &= [\![e]\!]_{m'}(\iota_{\Gamma,m,m'}(\rho), \rho_x) \\
[\![\mathrm{H}]\!]_m(\rho)(q) &= \delta((), \overline{\mathrm{H}}_q) & [\![\mathrm{T}]\!]_m(\rho)(q) &= \delta((), \overline{\mathrm{T}}_q) \\
[\![\mathrm{S}]\!]_m(\rho)(q) &= \delta((), \overline{\mathrm{S}}_q) & [\![\mathrm{CX}]\!]_m(\rho)((q_1, q_2)) &= \delta((), \overline{\mathrm{CX}}_{q_1,q_2}) \\
[\![\mathrm{X}]\!]_m(\rho)(q) &= \delta((), \overline{\mathrm{X}}_q) & [\![\textsc{meas}]\!]_m(\rho)(q) &= \delta(\mathsf{False}, \overline{|0\rangle\langle 0|}_q) + \delta(\mathsf{True}, \overline{|1\rangle\langle 1|}_q) \\
[\![\textsc{alloc}]\!]_m(\rho)(u) &= \delta(m, \overline{V}) & \text{where } V(v) &= v \otimes |0\rangle
\end{aligned}$$

where ρ_x refers to the x^{th} projection from the semantics of the context, and we recall that for a linear map f we define:

$$\overline{f} = \rho \mapsto f^\dagger \circ \rho \circ f$$

and then use subscripts to refer to the qubits of $\mathcal{H}^m$ that this channel should target.

Expressions $\Gamma \vdash^e_n e : X$ have semantics given by (continuous) functions:

$$[\![e]\!]_m \in \mathbf{DCPO}([\![\Gamma]\!]_m, \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n}), [\![X]\!]_{m+n}))$$

The semantics for each constructor of expressions are given by the following composites:

$$\begin{aligned}
[\![\mathsf{return}\, v]\!]_m &= [\![\Gamma]\!]_m \xrightarrow{[\![v]\!]_m} [\![X]\!]_m \xrightarrow{\eta_{\mathcal{B}(\mathcal{H}^m)X}} \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m) \\
[\![\mathsf{let}\, x = e_1 \,\mathsf{in}\, e_2]\!]_m &= [\![\Gamma]\!]_m \\
&\quad \Big\downarrow \langle \iota_{\Gamma,m,m+n_1}, [\![e_1]\!]_m \rangle \\
&\quad [\![\Gamma]\!]_{m+n_1} \times \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![X]\!]_{m+n_1}) \\
&\quad \Big\downarrow \tau_{[\![\Gamma]\!]_{m+n_1}, \mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![X]\!]_{m+n_1}} \\
&\quad \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![\Gamma]\!]_{m+n_1} \times [\![X]\!]_{m+n_1}) \\
&\quad \Big\downarrow [\![e_2]\!]^\sharp_{m+n_1} \\
&\quad \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1+n_2}), [\![Y]\!]_{m+n_1+n_2})
\end{aligned}$$

where $\Gamma \vdash^{e}_{n_1} e_1 : X$ and $\Gamma, x : X \vdash^{e}_{n_2} e_2 : Y$

$$
\begin{aligned}
[\![e_1; e_2]\!]_m &= [\![\Gamma]\!]_m \\
&\downarrow \langle \iota_{\Gamma,m,m+n_1}, [\![e_1]\!]_m \rangle \\
&[\![\Gamma]\!]_{m+n_1} \times \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![X]\!]_{m+n_1}) \\
&\downarrow \tau_{[\![\Gamma]\!]_{m+n_1}, \mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![X]\!]_{m+n_1}} \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![\Gamma]\!]_{m+n_1} \times [\![X]\!]_{m+n_1}) \\
&\downarrow \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), \pi_1) \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![\Gamma]\!]_{m+n_1}) \\
&\downarrow [\![e_2]\!]^{\sharp}_{m+n_1} \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1+n_2}), [\![Y]\!]_{m+n_1+n_2})
\end{aligned}
$$

where $\Gamma \vdash^{e}_{n_1} e_1 : X$ and $\Gamma, x : X \vdash^{e}_{n_2} e_2 : Y$

$$
\begin{aligned}
[\![v_1(v_2)]\!]_m &= [\![\Gamma]\!]_m \\
&\downarrow \langle [\![v_1]\!]_m, [\![v_2]\!]_m \rangle \\
&(\mathbf{DCPO}([\![X]\!]_m, \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n_1}), [\![Y]\!]_{m+n_1}))) \times [\![X]\!]_m \\
&\downarrow \mathrm{eval} \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n}), [\![Y]\!]_{m+n})
\end{aligned}
$$

where $\Gamma \vdash^{v} v_1 : X \to_n Y$

$$
\begin{aligned}
[\![\mathsf{if}\, v \,\mathsf{then}\, e_1 \,\mathsf{else}\, e_2]\!]_m &= [\![\Gamma]\!]_m \\
&\downarrow \langle [\![v]\!]_m, \mathrm{id} \rangle \\
&[\![\mathsf{bool}]\!]_m \times [\![\Gamma]\!]_m \\
&\downarrow \simeq \\
&[\![\Gamma]\!]_m + [\![\Gamma]\!]_m \\
&\downarrow [[\![e_1]\!]_m, [\![e_2]\!]_m] \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^{m+n}), [\![X]\!]_{m+n}) \\
[\![\mathsf{fix}\, f]\!]_m &= [\![\Gamma]\!]_m \\
&\downarrow [\![f]\!]_m \\
&\mathbf{DCPO}((1 \to \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)), \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)) \\
&\downarrow \simeq \\
&\mathbf{DCPO}(\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m), \mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)) \\
&\downarrow \mathsf{fix} \\
&\mathcal{Q}(\mathcal{B}(\mathcal{H}^m), \mathcal{B}(\mathcal{H}^m), [\![X]\!]_m)
\end{aligned}
$$

where $\langle f, g\rangle$ is the universal map into the product, and $[f, g]$ is the universal map out of the sum.

B Finite Quantum Instruments are a Strong Parametrised Monad

In this section we show that the finite quantum instruments used in Sections 3.3 and 3.4 actually form a monad.

Definition 35 (Parameterised Quantum Instrument Monad)**.** We define the $\mathbf{W^*}^{\mathrm{op}}$-parameterised quantum instrument monad Q as follows.

- Let $\mathscr{A}$ and $\mathscr{B}$ be W^*-algebras and X be a set. Then elements $\xi \in Q(\mathscr{A}, \mathscr{B}, X)$ are given by a collection $\{\xi_x \in \mathbf{W^*}(\mathscr{B}, \mathscr{A}) \mid x \in X\}$ such that:
  - $\mathrm{Supp}(\xi) \coloneqq \{x \in X \mid \xi_x \neq 0\}$ is finite.
  - $\sum_{x\in X} \xi_x$ is subunital.
- To show that Q is a functor $\mathbf{W^*} \times \mathbf{W^*}^{\mathrm{op}} \times \mathbf{Set} \to \mathbf{Set}$, we let $\phi : \mathbf{W^*}(\mathscr{A}, \mathscr{B})$, $\psi : \mathbf{W^*}(\mathscr{B}', \mathscr{A}')$, and $f : \mathbf{Set}(X, Y)$, and define $Q(\phi, \psi, f) : Q(\mathscr{A}, \mathscr{A}', X) \to Q(\mathscr{B}, \mathscr{B}', Y)$ by:

$$Q(\phi, \psi, f)(\xi)_y = \sum_{x\in f^{-1}(y)} \phi \circ \xi_x \circ \psi$$

  for $\xi \in Q(\mathscr{A}, \mathscr{A}', X)$ and $y \in Y$.
- We define the unit $\eta_{\mathscr{A}X} : X \to Q(\mathscr{A}, \mathscr{A}, X)$ by:

$$\eta_{\mathscr{A}X}(x) = \delta(x, \mathrm{id}_{\mathscr{A}})$$

  for $x \in X$, recalling that δ is defined as the *Dirac quantum instrument*.
- Let the multiplication $\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X} : Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X)) \to Q(\mathscr{A}, \mathscr{C}, X)$ be defined by:

$$\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_x = \sum_{\xi\in Q(\mathscr{B},\mathscr{C},X)} \Xi_\xi \circ \xi_x$$

  for $x \in X$.
- For completeness, we explicitly give the strength $\tau_{X,\mathscr{A},\mathscr{B},Y} : X \times Q(\mathscr{A}, \mathscr{B}, Y) \to Q(\mathscr{A}, \mathscr{B}, X \times Y)$ by:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}(x, \xi)_{(x',y)} = \delta(x, \xi_y)_{x'}$$

  for each $x, x' \in X$ and $y \in Y$.

We use the multiplication instead of the Kleisli extension above to align more closely with the laws of a parameterised monad. The extension of a map $f : X \to Q(\mathscr{B}, \mathscr{C}, Y)$ is given by:

$$f^{\#}_{\mathscr{A}} : Q(\mathscr{A}, \mathscr{B}, X) \to Q(\mathscr{A}, \mathscr{C}, Y)$$
$$f^{\#}_{\mathscr{A}} = \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y} \circ Q(\mathscr{A}, \mathscr{B}, f)$$

expanding out these definitions, we get:

$$\begin{aligned} f^{\#}_{\mathscr{A}}(\xi)_y &= \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(Q(\mathscr{A},\mathscr{B},f)(\xi))_y \\ &= \sum_{\chi \in Q(\mathscr{B},\mathscr{C},Y)} \left(Q(\mathscr{A},\mathscr{B},f)(\xi)_\chi\right) \circ \chi_y \\ &= \sum_{\chi \in Q(\mathscr{B},\mathscr{C},Y)} \left(\sum_{x \in f^{-1}(\chi)} \xi_x \right) \circ \chi_y \\ &= \sum_{\substack{\chi \in Q(\mathscr{B},\mathscr{C},Y) \\ x \in f^{-1}(\chi)}} \xi_x \circ \chi_y \\ &= \sum_{x \in X} \xi_x \circ f(x)_y \end{aligned}$$

for $\xi \in Q(\mathscr{A},\mathscr{B},X)$ and $y \in Y$, aligning with the extension given in the previous section.

Lemma 36. *If $x \in X$ and $\Phi \in \mathbf{W}^*(\mathscr{B},\mathscr{A})$, then the Dirac quantum instrument $\delta(x,\Phi) \in Q(\mathscr{A},\mathscr{B},X)$ is well-defined. Further, we have:*

$$Q(\phi,\psi,f)(\delta(x,\Phi)) = \delta(f(x), \phi \circ \Phi \circ \psi)$$

for $f : X \to Y$, $\phi \in \mathbf{W}^*(\mathscr{A},\mathscr{A}')$, and $\psi \in \mathbf{W}^*(\mathscr{B}',\mathscr{B})$.

Proof. The support of $\delta(x,\Phi)$ is $\{x\}$ and so is finite. The sum of all the components is Φ, which is subunital by definition. Hence $\delta(x,\Phi) \in Q(\mathscr{A},\mathscr{B},X)$. Let f, ϕ, and ψ be as in the statement of the lemma. Then:

$$\begin{aligned} Q(\phi,\psi,f)(\delta(x,\Phi))_y &= \sum_{x' \in f^{-1}(y)} \phi \circ \delta(x,\Phi)_{x'} \circ \psi \\ &= \begin{cases} \phi \circ \Phi \circ \psi & \text{if } x \in f^{-1}(y) \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \phi \circ \Phi \circ \psi & \text{if } f(x) = y \\ 0 & \text{otherwise} \end{cases} \\ &= \delta(f(x), \phi \circ \Phi \circ \psi)_y \end{aligned}$$

for all y, and so the stated equality holds. □

Lemma 37. *Q is well-defined and a functor from $\mathbf{W}^* \times \mathbf{W}^{*op} \times \mathbf{Set} \to \mathbf{Set}$.*

Proof. Firstly,

$$Q(\mathrm{id}_{\mathscr{A}}, \mathrm{id}_{\mathscr{B}}, \mathrm{id}_X)(\xi)_x = \sum_{x' \in \mathrm{id}_X^{-1}(x)} \mathrm{id}_{\mathscr{A}} \circ \xi_x \circ \mathrm{id}_{\mathscr{B}} = \xi_x$$

and so $Q(\mathrm{id}_{\mathscr{A}}, \mathrm{id}_{\mathscr{B}}, \mathrm{id}_X) = \mathrm{id}_{Q(\mathscr{A},\mathscr{B},X)}$. Supposing $\phi : \mathbf{W}^*(\mathscr{A},\mathscr{B})$, $\psi : \mathbf{W}^*(\mathscr{B}',\mathscr{A}')$, and $f : X \to Y$, we must show that $Q(\phi,\psi,f)$ is well defined. Specifically, for $\xi \in Q(\mathscr{A},\mathscr{A}',X)$, we have that:

$$\begin{aligned} \mathrm{Supp}(Q(\phi,\psi,f)(\xi)) &= \left\{ y \in Y \mid \textstyle\sum_{x \in f^{-1}(y)} \phi \circ \xi_x \circ \psi \neq 0 \right\} \\ &\subseteq \left\{ y \in Y \mid \textstyle\sum_{x \in f^{-1}(y)} \xi_x \neq 0 \right\} \\ &\subseteq f(\mathrm{Supp}(\xi)) \end{aligned}$$

and so the support is finite. Then:

$$\begin{aligned}\sum_{y\in Y} Q(\phi,\psi,f)(\xi)_y &= \sum_{\substack{y\in Y\\ x\in f^{-1}(y)}} \phi\circ\xi_x\circ\psi\\ &= \sum_{x\in X} \phi\circ\xi_x\circ\psi\\ &= \phi\circ\left(\sum_{x\in X}\xi_i\right)\circ\psi\end{aligned}$$

which is a composition of subunital maps and so is subunital. Hence, $Q(\phi,\psi,f)(\xi)\in Q(\mathscr{B},\mathscr{B}',Y)$ and so $Q(\phi,\psi,f)$ is well-defined.

If we further suppose that we have $\phi':\mathbf{W}^*(\mathscr{B},\mathscr{C})$, $\psi':\mathbf{W}^*(\mathscr{C}',\mathscr{B}')$, and $g:Y\to Z$, then:

$$\begin{aligned}Q(\phi',\psi',g)(Q(\phi,\psi,f)(\xi))_z &= \sum_{y\in g^{-1}(z)} \phi'\circ Q(\phi,\psi,f)(\xi)_y\circ\psi'\\ &= \sum_{y\in g^{-1}(z)} \phi'\circ\left(\sum_{x\in f^{-1}(y)}\phi\circ\xi_x\circ\psi\right)\circ\psi'\\ &= \sum_{\substack{y\in g^{-1}(z)\\ x\in f^{-1}(y)}} \phi'\circ\phi\circ\xi_x\circ\psi\circ\psi'\\ &= \sum_{x\in (g\circ f)^{-1}(z)} \phi'\circ\phi\circ\xi_x\circ\psi\circ\psi'\\ &= Q(\phi'\circ\phi,\psi\circ\psi',g\circ f)(\xi)_z\end{aligned}$$

and so Q is a functor. □

Lemma 38. *The unit η is a natural transformation.*

Proof. As η is a Dirac quantum instrument it is well-defined by Lemma 36.

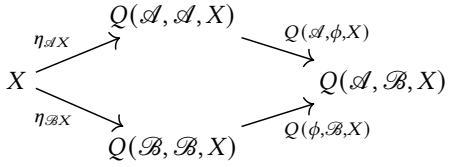
To show naturality of η, we need naturality in X and dinaturality in $\mathscr{A}$. The first requires that the following square commutes for all $\mathscr{A}$:

$$\begin{array}{ccc} X & \xrightarrow{\quad f\quad} & Y\\ {\scriptstyle \eta_{\mathscr{A}X}}\downarrow & & \downarrow{\scriptstyle \eta_{\mathscr{A}Y}}\\ Q(\mathscr{A},\mathscr{A},X) & \xrightarrow{Q(\mathscr{A},\mathscr{A},f)} & Q(\mathscr{A},\mathscr{A},Y)\end{array}$$

By Lemma 36, it holds that

$$Q(\mathscr{A},\mathscr{A},f)(\eta_{\mathscr{A}X}(x)) = Q(\mathscr{A},\mathscr{A},f)(\delta(x,\mathrm{id}_{\mathscr{A}})) = \delta(f(x),\mathrm{id}_{\mathscr{A}}) = \eta_{\mathscr{A}Y}(f(x)).$$

Showing dinaturality in $\mathscr{A}$ amounts to proving the following diagram commutes for $\phi:\mathbf{W}^*(\mathscr{B},\mathscr{A})$:



Letting $x \in X$ and again using Lemma 36:

$$Q(\mathscr{A}, \phi, X)(\eta_{\mathscr{A}X}(x)) = Q(\mathscr{A}, \phi, X)(\delta(x, \mathrm{id}_{\mathscr{A}})) = \delta(x, \phi) = Q(\phi, \mathscr{B}, X)(\delta(x, \mathrm{id}_{\mathscr{B}})) = \eta_{\mathscr{B}X}(x),$$

and so η is natural as required. □

Lemma 39. *The multiplication μ is a natural transformation.*

Proof. For an instrument $\Xi \in Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X))$ and $x \in X$, we have that $\mu_{\mathscr{A},\mathscr{B},\mathscr{C}}(\Xi)_x \neq 0$ implies that there is $\xi \in \mathrm{Supp}(\Xi)$ with $x \in \mathrm{Supp}(\xi)$. Therefore,

$$\mathrm{Supp}\left(\mu_{\mathscr{A},\mathscr{B},\mathscr{C}}(\Xi)\right) \subseteq \bigcup_{\xi \in \mathrm{Supp}(\Xi)} \mathrm{Supp}(\xi),$$

with the right-hand side being a finite union of finite sets. We then note that

$$\sum_{x \in X} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_x = \sum_{\substack{x \in X \\ \xi \in Q(\mathscr{B},\mathscr{C},X)}} \Xi_\xi \circ \xi_x = \sum_{\xi \in Q(\mathscr{B},\mathscr{C},X)} \Xi_\xi \circ \Big(\sum_{x \in X} \xi_x\Big),$$

and hence,

$$\sum_{x \in X} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_x(1_{\mathscr{C}}) = \sum_{\xi \in Q(\mathscr{B},\mathscr{C},X)} \Xi_\xi\Big(\sum_{x \in X} \xi_x(1_{\mathscr{C}})\Big) \leq \sum_{\xi \in Q(\mathscr{B},\mathscr{C},X)} \Xi_\xi(1_{\mathscr{B}}) \leq 1_{\mathscr{A}},$$

where the second equality holds as $\sum_{x \in X} \xi_x$ is subunital for each ξ and the third holds as the sum of the components of Ξ is subunital.

We now require naturality in $\mathscr{A}, \mathscr{C}$, and X, with dinaturality in $\mathscr{B}$. We treat each case separately:

- Naturality in X: We must show the following square commutes:

$$\begin{array}{ccc} Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X)) & \xrightarrow{Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},f))} & Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, Y)) \\ \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}} & & \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}} \\ Q(\mathscr{A}, \mathscr{C}, X) & \xrightarrow{\qquad Q(\mathscr{A},\mathscr{C},f) \qquad} & Q(\mathscr{A}, \mathscr{C}, Y) \end{array}$$

  Take $\Xi \in Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X))$. Then for $y \in Y$ we have:

$$\begin{aligned} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, f))(\Xi))_y &= \sum_{\xi \in Q(\mathscr{B},\mathscr{C},Y)} Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, f))(\Xi)_\phi \circ \xi_y \\ &= \sum_{\substack{\xi \in Q(\mathscr{B},\mathscr{C},Y) \\ \xi' \in Q(\mathscr{B},\mathscr{C},f)^{-1}(\xi)}} \Xi_{\xi'} \circ \xi'_y \\ &= \sum_{\xi' \in Q(\mathscr{B},\mathscr{C},X)} \Xi_{\xi'} \circ Q(\mathscr{B}, \mathscr{C}, f)(\xi')_y \\ &= \sum_{\substack{x \in f^{-1}(y) \\ \xi' \in Q(\mathscr{B},\mathscr{C},X)}} \Xi_{\xi'} \circ \xi'_x \\ &= \sum_{x \in f^{-1}(y)} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi)_x \\ &= Q(\mathscr{A}, \mathscr{C}, f)(\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(\Xi))_y \end{aligned}$$

  and so the square commutes.

- Naturality in $\mathscr{A}$ and $\mathscr{C}$: Suppose $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{A}')$ and $\psi \in \mathbf{W}^*(\mathscr{C}', \mathscr{C})$, then the following square must commute:

$$\begin{array}{ccc} Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X)) & \xrightarrow{Q(\phi, \mathscr{B}, Q(\mathscr{B}, \psi, X))} & Q(\mathscr{A}', \mathscr{B}, Q(\mathscr{B}, \mathscr{C}', X)) \\ \downarrow{\scriptstyle \mu_{\mathscr{A}, \mathscr{B}, \mathscr{C}, X}} & & \downarrow{\scriptstyle \mu_{\mathscr{A}', \mathscr{B}, \mathscr{C}', X}} \\ Q(\mathscr{A}, \mathscr{C}, X) & \xrightarrow{Q(\phi, \psi, X)} & Q(\mathscr{A}', \mathscr{C}', X) \end{array}$$

Again taking $\Xi \in Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X))$ and $x \in X$ we have:

$$\begin{aligned} \mu_{\mathscr{A}', \mathscr{B}, \mathscr{C}', X}(Q(\phi, \mathscr{B}, Q(\mathscr{B}, \psi, X))(\Xi)) &= \sum_{\xi' \in Q(\mathscr{B}, \mathscr{C}', X)} Q(\phi, \mathscr{B}, Q(\mathscr{B}, \psi, X))(\Xi)_{\xi'} \circ \xi'_x \\ &= \sum_{\xi' \in Q(\mathscr{B}, \mathscr{C}', X)} \Big(\sum_{\xi \in Q(\mathscr{B}, \psi, X)^{-1}(\xi')} \phi \circ \Xi_\xi \Big) \circ \xi'_x \\ &= \sum_{\substack{\xi' \in Q(\mathscr{B}, \mathscr{C}', X) \\ \xi \in Q(\mathscr{B}, \psi, X)^{-1}(\xi')}} \phi \circ \Xi_\xi \circ \xi'_x \\ &= \sum_{\xi \in Q(\mathscr{B}, \mathscr{C}, X)} \phi \circ \Xi_\xi \circ \xi_x \circ \psi \\ &= \phi \circ \Big(\sum_{\xi \in Q(\mathscr{B}, \mathscr{C}, X)} \Xi_\xi \circ \xi_x \Big) \circ \psi \\ &= \phi \circ \mu_{\mathscr{A}, \mathscr{B}, \mathscr{C}, X}(\Xi)_x \circ \psi \\ &= Q(\phi, \psi, X)(\mu_{\mathscr{A}, \mathscr{B}, \mathscr{C}, X}(\Xi))_x \end{aligned}$$

- Dinaturality in $\mathscr{B}$. Suppose $\phi \in \mathbf{W}^*(\mathscr{B}', \mathscr{B})$. Dinaturality is the commutativity of the following square:

$$\begin{array}{ccccc} & & Q(\mathscr{A}, \mathscr{B}', Q(\mathscr{B}', \mathscr{C}, X)) & & \\ & \nearrow{\scriptstyle Q(\mathscr{A}, \phi, Q(\mathscr{B}', \mathscr{C}, X))} & & \searrow{\scriptstyle \mu_{\mathscr{A}, \mathscr{B}', \mathscr{C}, X}} & \\ Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}', \mathscr{C}, X)) & & & & Q(\mathscr{A}, \mathscr{C}, X) \\ & \searrow{\scriptstyle Q(\mathscr{A}, \mathscr{B}, Q(\phi, \mathscr{C}, X))} & & \nearrow{\scriptstyle \mu_{\mathscr{A}, \mathscr{B}, \mathscr{C}, X}} & \\ & & Q(\mathscr{A}, \mathscr{B}, Q(\mathscr{B}, \mathscr{C}, X)) & & \end{array}$$

With $\Xi \in Q(\mathscr{A}, \mathscr{B}', Q(\mathscr{B}', \mathscr{C}, X))$ and $x \in X$:

$$\begin{aligned}
\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}(Q(\mathscr{A},\mathscr{B},Q(\phi,\mathscr{C},X))(\Xi))_x &= \sum_{\xi \in Q(\mathscr{B},\mathscr{C},X)} Q(\mathscr{A},\mathscr{B},Q(\phi,\mathscr{C},X))(\Xi)_\xi \circ \xi_x \\
&= \sum_{\xi \in Q(\mathscr{B},\mathscr{C},X)} \Big(\sum_{\xi' \in Q(\phi,\mathscr{C},X)^{-1}(\xi)} \Xi_{\xi'} \Big) \circ \xi_x \\
&= \sum_{\substack{\xi \in Q(\mathscr{B},\mathscr{C},X) \\ \xi' \in Q(\phi,\mathscr{C},X)^{-1}(\xi)}} \Xi_{\xi'} \circ \xi_x \\
&= \sum_{\xi' \in Q(\mathscr{B}',\mathscr{C},X)} \Xi_{\xi'} \circ \phi \circ \xi'_x \\
&= \sum_{\xi' \in Q(\mathscr{B}',\mathscr{C},X)} Q(\mathscr{A},\phi,Q(\mathscr{B}',\mathscr{C},X))(\Xi)_{\xi'} \circ \xi'_x \\
&= \mu_{\mathscr{A},\mathscr{B}',\mathscr{C},X}(Q(\mathscr{A},\phi,Q(\mathscr{B}',\mathscr{C},X))(\Xi))_x
\end{aligned}$$

Hence μ is natural. □

Lemma 40. *The strength τ is a natural transformation.*

Proof. We first note that $\mathrm{Supp}(\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)) = \{(x,y) \mid y \in \mathrm{Supp}(\xi)\}$, and so is finite. The sum of components of $\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)$ is equal to the sum of components of ξ so is subunital. Hence, $\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)$ is well-defined.

Now suppose $\phi \in \mathbf{W}^*(\mathscr{A},\mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}',\mathscr{B})$, and $g \in \mathbf{Set}(Y,Y')$. Then:

$$\begin{aligned}
Q(\phi,\psi,\mathrm{id}_X \times g)(\tau_{X,\mathscr{A},\mathscr{B},Y'}(x,\xi))_{(x',y)} &= \sum_{y' \in g^{-1}(y)} \phi \circ \tau_{X,\mathscr{A},\mathscr{B},Y'}(x,\xi)_{(x',y')} \circ \psi \\
&= \sum_{y' \in g^{-1}(y)} \phi \circ \delta(x,\xi_{y'})_{x'} \circ \psi \\
&= \delta(x, \sum_{y' \in g^{-1}(y)} \phi \circ \xi_{y'} \circ \psi)_{x'} \\
&= \delta(x, Q(\phi,\psi,g)(\xi)_y)_{x'} \\
&= \tau_{X,\mathscr{A}',\mathscr{B}',Y}(x, Q(\phi,\psi,g)(\xi))_{(x',y)}
\end{aligned}$$

and for $f \in \mathbf{Set}(X,X')$:

$$\begin{aligned}
Q(\mathscr{A},\mathscr{B},f \times \mathrm{id}_Y)(\tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi))_{(x',y)} &= \sum_{x'' \in f^{-1}(x')} \tau_{X,\mathscr{A},\mathscr{B},Y}(x,\xi)_{(x'',y)} \\
&= \sum_{x'' \in f^{-1}(x')} \delta(x,\xi_y)_{x''} \\
&= Q(\mathscr{A},\mathscr{B},f)(\delta(x,\xi_y))_{x'} \\
&= \delta(f(x),\xi_y)_{x'} \\
&= \tau_{X',\mathscr{A},\mathscr{B},Y}(f(x),\xi)_{(x',y)}
\end{aligned}$$

where the second to last line is by Lemma 36 and so τ is a natural transformation. □

Theorem 41. *(Q,η,μ,τ) is a $\mathbf{W}^{*^{op}}$-parameterised monad.*

Proof. We begin with the monad laws for unitality. We must have $\mu_{\mathscr{A},\mathscr{B},\mathscr{B},X} \circ Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X}) = \mathrm{id}_{Q(\mathscr{A},\mathscr{B},X)}$. Taking $\xi \in Q(\mathscr{A},\mathscr{B},X)$ and $x \in X$:

$$\begin{aligned}
\mu_{\mathscr{A},\mathscr{B},\mathscr{B},X}(Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X})(\xi))_x &= \sum_{\xi' \in Q(\mathscr{B},\mathscr{B},X)} Q(\mathscr{A},\mathscr{B},\eta_{\mathscr{A}X})(\xi)_{\xi'} \circ \xi'_x \\
&= \sum_{\substack{\xi' \in Q(\mathscr{B},\mathscr{B},X) \\ x' \in \eta^{-1}_{\mathscr{A}X}(\xi')}} \xi_{x'} \circ \xi'_x \\
&= \sum_{x' \in X} \xi_{x'} \circ \eta_{\mathscr{A}X}(x')_x \\
&= \xi_x
\end{aligned}$$

Similarly we must have $\mu_{\mathscr{A},\mathscr{A},\mathscr{B},X} \circ \eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B},X)} = \mathrm{id}_{Q(\mathscr{A},\mathscr{B},X)}$. Again for all ξ and x we have:

$$\begin{aligned}
\mu_{\mathscr{A},\mathscr{A},\mathscr{B},X}(\eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B},X)}(\xi))_x &= \sum_{\xi' \in Q(\mathscr{A},\mathscr{B},X)} \eta_{\mathscr{A}Q(\mathscr{A},\mathscr{B},X)}(\xi)_{\xi'} \circ \xi'_x \\
&= \sum_{\xi' \in Q(\mathscr{B},\mathscr{B},X)} \begin{cases} \mathrm{id}_{\mathscr{A}} \circ \xi'_x & \text{if } \xi = \xi' \\ 0 & \text{otherwise} \end{cases} \\
&= \xi_x
\end{aligned}$$

For the associativity monad law, we need the following diagram to commute:

$$\begin{array}{ccc}
Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X))) & \xrightarrow{\mu_{\mathscr{A},\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)}} & Q(\mathscr{A},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)) \\
\downarrow{\scriptstyle Q(\mathscr{A},\mathscr{B},\mu_{\mathscr{B},\mathscr{C},\mathscr{D},X})} & & \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{C},\mathscr{D},X}} \\
Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{D},X)) & \xrightarrow{\mu_{\mathscr{A},\mathscr{B},\mathscr{D},X}} & Q(\mathscr{A},\mathscr{D},X)
\end{array}$$

Take $\aleph \in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)))$ and $x \in X$. Then evaluating the bottom left of the square gives:

$$\begin{aligned}
\mu_{\mathscr{A},\mathscr{B},\mathscr{D},X}(Q(\mathscr{A},\mathscr{B},\mu_{\mathscr{B},\mathscr{C},\mathscr{D},X})(\aleph))_x &= \sum_{\xi \in Q(\mathscr{B},\mathscr{D},X)} Q(\mathscr{A},\mathscr{B},\mu_{\mathscr{B},\mathscr{C},\mathscr{D},X})(\aleph)_\xi \circ \xi_x \\
&= \sum_{\xi \in Q(\mathscr{B},\mathscr{D},X)} \sum_{\Xi \in \mu^{-1}_{\mathscr{B},\mathscr{C},\mathscr{D},X}(\xi)} \aleph_\Xi \circ \xi_x \\
&= \sum_{\substack{\xi \in Q(\mathscr{B},\mathscr{D},X) \\ \Xi \in \mu^{-1}_{\mathscr{B},\mathscr{C},\mathscr{D},X}(\xi)}} \aleph_\Xi \circ \xi_x \\
&= \sum_{\Xi \in Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X))} \aleph_\Xi \circ \mu_{\mathscr{B},\mathscr{C},\mathscr{D},X}(\Xi)_x \\
&= \sum_{\Xi \in Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X))} \aleph_\Xi \circ \Big(\sum_{\xi \in Q(X)} \Xi_\xi \circ \xi_x \Big) \\
&= \sum_{\substack{\Xi \in Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)) \\ \xi \in Q(\mathscr{C},\mathscr{D},X)}} \aleph_\Xi \circ \Xi_\xi \circ \xi_x
\end{aligned}$$

and evaluating the top right gives:

$$\begin{aligned}\mu_{\mathscr{A},\mathscr{C},\mathscr{D},X}(\mu_{\mathscr{A},\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)}(\aleph))_x &= \sum_{\xi\in Q(\mathscr{C},\mathscr{D},X)} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)}(\aleph)_\xi \circ \xi_x \\ &= \sum_{\xi\in Q(\mathscr{C},\mathscr{D},X)} \left(\sum_{\Xi\in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X))} \aleph_\Xi \circ \Xi_\xi \right) \circ \xi_x \\ &= \sum_{\substack{\Xi\in Q(\mathscr{B},\mathscr{C},Q(\mathscr{C},\mathscr{D},X)) \\ \xi\in Q(\mathscr{C},\mathscr{D},X)}} \aleph_\Xi \circ \Xi_\xi \circ \xi_x\end{aligned}$$

and so the square commutes as required.

For strength, we first show that it interacts well with $1 = \{*\}$:

$$Q(\mathscr{A},\mathscr{B},\lambda_X)\circ\tau_{1,\mathscr{A},\mathscr{B},X} = \lambda_{Q(\mathscr{A},\mathscr{B},X)}$$

Where $\lambda_X : 1\times X \to X$ is the left unitor isomorphism. For $\xi \in Q(\mathscr{A},\mathscr{B},X)$ and $x \in X$:

$$\begin{aligned}Q(\mathscr{A},\mathscr{B},\lambda_X)(\tau_{1,\mathscr{A},\mathscr{B},X}(*,\xi))_x &= \sum_{x\in\lambda_X^{-1}(x)} \tau_{1,\mathscr{A},\mathscr{B},X}(*,\xi)_x \\ &= \tau_{1,\mathscr{A},\mathscr{B},X}(*,\xi)_{(*,x)} \\ &= \xi_x \\ &= \lambda_{Q(\mathscr{A},\mathscr{B},X)}(*,\xi)_x\end{aligned}$$

For compatibility with the associator isomorphism $\alpha_{X,Y,Z} : (X\times Y)\times Z \to X\times(Y\times Z)$, we need the following to commute:

$$\begin{array}{ccccc}(X\times Y)\times Q(\mathscr{A},\mathscr{B},Z) & & \xrightarrow{\quad\tau_{X\times Y,\mathscr{A},\mathscr{B},Z}\quad} & & Q(\mathscr{A},\mathscr{B},(X\times Y)\times Z) \\ \downarrow{\scriptstyle \alpha_{X,Y,Q(\mathscr{A},\mathscr{B},Z)}} & & & & \downarrow{\scriptstyle Q(\mathscr{A},\mathscr{B},\alpha_{X,Y,Z})} \\ X\times(Y\times Q(\mathscr{A},\mathscr{B},Z)) & \xrightarrow{\mathrm{id}_X\times\tau_{Y,\mathscr{A},\mathscr{B},Z}} & X\times Q(\mathscr{A},\mathscr{B},Y\times Z) & \xrightarrow{\tau_{X,\mathscr{A},\mathscr{B},Y\times Z}} & Q(\mathscr{A},\mathscr{B},X\times(Y\times Z))\end{array}$$

With $\xi \in Q(\mathscr{A},\mathscr{B},Z)$, $x,x' \in X$, $y,y' \in Y$, and $z \in Z$:

$$\begin{aligned}&\tau_{X,\mathscr{A},\mathscr{B},Y\times Z}(\mathrm{id}_X\times\tau_{Y,\mathscr{A},\mathscr{B},Z}(\alpha_{X,Y,Q(\mathscr{A},\mathscr{B},Z)}((x',y'),\xi)))_{(x,(y,z))} \\ &= \tau_{X,\mathscr{A},\mathscr{B},Y\times Z}(x',\tau_{Y,\mathscr{A},\mathscr{B},Z}(y',\xi))_{(x,(y,z))} \\ &= \begin{cases}\tau_{Y,\mathscr{A},\mathscr{B},Z}(y',\xi)_{(y,z)} & \text{if } x = x' \\ 0 & \text{otherwise}\end{cases} \\ &= \begin{cases}\xi_z & \text{if } x = x' \text{ and } y = y' \\ 0 & \text{otherwise}\end{cases} \\ &= \tau_{X\times Y,\mathscr{A},\mathscr{B},Z}((x',y'),\xi)_{((x,y),z)} \\ &= Q(\mathscr{A},\mathscr{B},\alpha_{X,Y,Z})(\tau_{X\times Y,\mathscr{A},\mathscr{B},Z}((x',y'),\xi))_{(x,(y,z))}\end{aligned}$$

For compatibility with the unit η:

$$\tau_{X,\mathscr{A},\mathscr{A},Y}\circ(\mathrm{id}_X\times\eta_{\mathscr{A}Y}) = \eta_{\mathscr{A}X\times Y}$$

Taking $x, x' \in X$, and $y, y' \in Y$:

$$\begin{aligned}\tau_{X,\mathscr{A},\mathscr{A},Y}(x, \eta_{\mathscr{A}Y}(y))_{(x',y')} &= \begin{cases}\eta_{\mathscr{A},\mathscr{B},Y}(y)_{y'} & \text{if } x = x' \\ 0 & \text{otherwise}\end{cases} \\ &= \begin{cases}\mathrm{id}_{\mathscr{A}} & \text{if } x = x' \text{ and } y = y' \\ 0 & \text{otherwise}\end{cases} \\ &= \eta_{\mathscr{A}X\times Y}(x, y)_{(x',y')}\end{aligned}$$

Finally for compatibility with μ the following must commute:

$$\begin{array}{ccccc} X \times Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)) & \xrightarrow{\tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}} & Q(\mathscr{A},\mathscr{B},X\times Q(\mathscr{B},\mathscr{C},Y)) & \xrightarrow{Q(\mathscr{A},\mathscr{B},\tau_{X,\mathscr{B},\mathscr{C},Y})} & Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},X\times Y)) \\ \downarrow{\scriptstyle \mathrm{id}_X \times \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}} & & & & \downarrow{\scriptstyle \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X\times Y}} \\ X \times Q(\mathscr{A},\mathscr{C},Y) & & \xrightarrow{\qquad\qquad\tau_{X,\mathscr{A},\mathscr{C},Y}\qquad\qquad} & & Q(\mathscr{A},\mathscr{C},X\times Y) \end{array}$$

With $x', x \in X$, $y \in Y$, and $\Xi \in Q(\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y))$:

$$\begin{aligned}&\mu_{\mathscr{A},\mathscr{B},\mathscr{C},X\times Y}(Q(\mathscr{A},\mathscr{B},\tau_{X,\mathscr{B},\mathscr{C},Y})(\tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}(x',\Xi)))_{(x,y)} \\ &= \sum_{\xi\in Q(\mathscr{B},\mathscr{C},X\times Y)} Q(\mathscr{A},\mathscr{B},\tau_{X,\mathscr{B},\mathscr{C},Y})(\tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}(x',\Xi))_{\xi} \circ \xi_{(x,y)} \\ &= \sum_{\substack{\xi\in Q(\mathscr{B},\mathscr{C},X\times Y) \\ (x'',\xi')\in\tau^{-1}_{X,\mathscr{B},\mathscr{C},Y}(\xi)}} \tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}(x',\Xi)_{(x'',\xi')} \circ \xi_{(x,y)} \\ &= \sum_{\substack{x''\in X \\ \xi'\in Q(\mathscr{B},\mathscr{C},Y)}} \tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}(x',\Xi)_{(x'',\xi')} \circ \tau_{X,\mathscr{B},\mathscr{C},Y}(x'',\xi')_{(x,y)} \\ &= \sum_{\xi'\in Q(\mathscr{B},\mathscr{C},Y)} \tau_{X,\mathscr{A},\mathscr{B},Q(\mathscr{B},\mathscr{C},Y)}(x',\Xi)_{(x,\xi')} \circ \xi'_y \\ &= \begin{cases}\sum_{\xi'\in Q(\mathscr{B},\mathscr{C},Y)} \Xi_{\xi'} \circ \xi'_y & \text{if } x = x' \\ 0 & \text{otherwise}\end{cases} \\ &= \begin{cases}\mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\Xi)_y & \text{if } x = x' \\ 0 & \text{otherwise}\end{cases} \\ &= \tau_{X,\mathscr{A},\mathscr{C},Y}(x', \mu_{\mathscr{A},\mathscr{B},\mathscr{C},Y}(\Xi))_{(x,y)}\end{aligned}$$

Therefore, (Q, η, μ, τ) is a parameterised strong monad. □

C Finite Quantum Orchestras

The collection of quantum orchestras introduced above is large, and it can be unwieldy to perform calculations or define operations on it, motivating the search for well-behaved subsets of this structure. We have already seen the Dirac quantum orchestra, which corresponds to a quantum operations which always returns the same classical value. A natural extension is to consider the quantum orchestras who have a finite number of possible outputs. We will see that these correspond closely to quantum instruments.

Definition 42 (Finite Quantum Orchestra)**.** Recall that $\delta(x, \phi)$ is a Dirac quantum orchestra, and let addition of quantum orchestras be given pointwise (when this is well defined). Then, a quantum

orchestra is *finite* if it takes the form:

$$\sum_i \delta(x_i, \phi_i)$$

where $x_0, \ldots, x_n \in X$ and $\phi_0, \ldots, \phi_n \in \mathbf{W}^*(\mathscr{B}, \mathscr{A})$ such that:

$$\phi_0(1) + \cdots + \phi_n(1) \leq 1$$

that is $\sum \phi_i$ is subunital. We write $\mathcal{Q}_{\mathrm{f}}(\mathscr{A}, \mathscr{B}, X)$ for the subset of $\mathcal{Q}(\mathscr{A}, \mathscr{B}, X)$ consiting of finite quantum orchestras.

The condition that $\sum_i \phi_i$ is subunital is clear necessary for being a quantum orchestra. Consider k which maps everything to the identity channel. Then:

$$\sum_i \delta(x_i, \phi_i)_k = \sum_i \phi_i \circ k(x_i) = \sum_i \phi_i$$

and so the right hand side must be subunital. It in fact turns out that this condition is sufficient to be a quantum orchestra, meaning all finite quantum orchestras take this form.

Proposition 43. *Let $x_0, \ldots, x_n$ and $\phi_0, \ldots, \phi_n$ be as in Definition 42. Then,*

$$\sum_i \delta(x_i, \phi_i) \in \mathcal{Q}_{\mathrm{f}}(\mathscr{A}, \mathscr{B}, X)$$

and so the finite quantum orchestras are exactly the sums of this form.

Proof. We must first show that the sum sends $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K}, \mathscr{B}))$ to an element of $\mathbf{W}^*(\mathscr{K}, \mathscr{A})$. Applying it to the unit 1 we get:

$$\begin{aligned}
\sum_i \delta(x_i, \phi_i)_k(1) &= \left(\sum_i \phi_i \circ k(x_i)\right)(1) \\
&= \sum_i \phi_i(k(x_i)(1)) \\
&\leq \sum_i \phi_i(1) && \text{as each } k(x_i) \text{ is subunital} \\
&\leq 1 && \text{by assumption on the } \phi_i
\end{aligned}$$

Continuity is clear from the continuity of the Dirac quantum orchestra and the continuity of addition. For subconvexity, let $\lambda, \rho \geq 0$ with $\lambda + \rho \leq 1$. Then for all k and k':

$$\sum_i \delta(x_i, \phi_i)_{\lambda k + \rho k'} = \sum_i \lambda \delta(x_i, \phi_i)_k + \rho \delta(x_i, \phi_i)_{k'} = \lambda \sum_i \delta(x_i, \phi_i)_k + \rho \sum_i \delta(x_i, \phi_i)_{k'}$$

Lastly, for compositionality, suppose $\chi \in \mathbf{W}^*(\mathscr{K}', \mathscr{K})$. Then:

$$\sum_i \delta(x_i, \phi_i)_{k(\_) \circ \chi} = \sum_i \delta(x_i, \phi_i) \circ \chi = \left(\sum_i \delta(x_i, \phi_i)\right) \circ \chi$$

and so the sum is a quantum orchestra. □

We now investigate the action of the monad operations on finite quantum instruments. We first consider the action of the functor. Let

$$\sum_i \delta(x_i, \phi_i)$$

be a finite quantum orchestra. Suppose $\phi \in \mathbf{W}^*(\mathscr{A}, \mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}', \mathscr{B})$ and $f : X \to Y$. Then:

$$\begin{aligned} \mathcal{Q}(\phi, \psi, f)\left(\sum_i \delta(x_i, \phi_i)\right)_k &= \phi \circ \sum_i \delta(x_i, \phi_i)_{\psi \circ k(f(\_))} \\ &= \sum_i \phi \circ \phi_i \circ \psi \circ k(f(x_i)) \\ &= \sum_i \delta(f(x_i), \phi \circ \phi_i \circ \psi)_k \end{aligned}$$

and so $\mathcal{Q}(\phi, \psi, f)$ preserves finiteness.

By definition, the unit of the monad always produces a finite quantum orchestra. The action of the strength of the monad is also clear. Suppose $x \in X$ and that:

$$\sum_i \delta(y_i, \phi_i)$$

is a finite quantum orchestras in $\mathcal{Q}(\mathscr{A}, \mathscr{B}, Y)$. Then:

$$\tau_{X,\mathscr{A},\mathscr{B},Y}\left(x, \sum_i \delta(y_i, \phi_i)\right)_k = \sum_i \delta(y_i, \phi_i)_{k(x,\_)} = \sum_i \phi_i \circ k(x, y_i) = \sum_i \delta(x, y_i)_k$$

We now move our attention to the multiplication. Let I be finite and ξ_i be a finite quantum orchestras for $i \in I$, each given by:

$$\xi_i = \sum_{j \in J_i} \delta(x_{ij}, \psi_{ij})$$

where $x \in X$ and $\psi_{ij} \in \mathbf{W}^*(\mathscr{C}, \mathscr{B})$. Further suppose:

$$\sum_{i \in I} \delta(\xi_i, \phi_i)$$

is a finite quantum orchestra. Then we have:

$$\begin{aligned} \mu_{\mathscr{A},\mathscr{B},\mathscr{C},X}\left(\sum_i \delta(\xi_i, \phi_i)\right)_k &= \sum_i \delta(\xi_i, \phi_i)_{\mathrm{ev}_k} \\ &= \sum_i \phi_i \circ \mathrm{ev}_k(\xi_i) \\ &= \sum_i \phi_i \circ \sum_{j \in J_i} \delta(x_{ij}, \psi_{ij})_k \\ &= \sum_i \sum_{j \in J_i} \phi_i \circ \psi_{ij} \circ k(x_{ij}) \end{aligned}$$

which produces a finite quantum orchestra (as the disjoint union of the J_i is finite).

Although we know the form that finite quantum orchestras take, it can be hard to utilise this to define operations on them, as their decompositions may not be unique. However, functions can be defined this way if they satisfy certain conditions.

Proposition 44. *Let $F : \mathbf{W}^* \to \mathbf{W}^*$ be a functor such that the maps on morphisms are additive. Then, there are functions $F^{\dagger}_{\mathscr{A},\mathscr{B},X} : \mathcal{Q}_{\mathrm{f}}(\mathscr{A}, \mathscr{B}, X) \to \mathcal{Q}_{\mathrm{f}}(F(\mathscr{A}), F(\mathscr{B}), X)$ given by:*

$$\sum_i \delta(x_i, \phi_i) \mapsto \sum_i \delta(x_i, F(\phi_i))$$

These maps form a natural transformation.

Proof. Suppose $\xi \in \mathcal{Q}_{\mathrm{f}}(\mathscr{A}, \mathscr{B}, X)$ and that there are decompositions:

$$\sum_{i \in I} \delta(x_i, \phi_i) = \xi = \sum_{j \in J} \delta(x'_j, \phi'_j)$$

Then we must show that:

$$\sum_{i \in I} \delta(x_i, F(\phi_i)) = \sum_{j \in J} \delta(x'_j, F(\phi'_j))$$

Now let S be the (finite) set $\{x_i \mid i \in I\} \cup \{x'_j \mid j \in J\}$. Then:

$$\begin{aligned}\sum_{i \in I} \delta(x_i, F(\phi_i)) &= \sum_{x \in S} \sum_{\substack{i \in I \\ x_i = x}} \delta(x, F(\phi_i)) \\ &= \sum_{x \in S} \delta\left(x, \sum_{\substack{i \in I \\ x_i = x}} F(\phi_i)\right) \\ &= \sum_{x \in S} \delta\left(x, F\left(\sum_{\substack{i \in I \\ x_i = x}} \phi_i\right)\right)\end{aligned}$$

Similarly we have:

$$\sum_{j \in J} \delta(x'_j, F(\phi'_j)) = \sum_{x \in S} \delta\left(x, F\left(\sum_{\substack{j \in J \\ x'_j = x}} \phi'_j\right)\right)$$

and so it is sufficient to show that for all $x \in S$ that:

$$\sum_{\substack{i \in I \\ x_i = x}} \phi_i = \sum_{\substack{j \in J \\ x'_j = x}} \phi'_j$$

Suppose for contradiction that this doesn't hold, and let $x \in S$ be maximal such that this equality doesn't hold. Then consider the set:

$$O = \bigcap_{\substack{s \in S \\ x \not\leq s}} s_{\not\leq} \quad \text{where} \quad s_{\not\leq} = \{y \in X \mid y \not\leq s\}$$

O is a Scott open subset of X: it is a finite intersection of sets of the form $s_{\not\leq}$ which are all upwards closed, as if $y \not\leq s$ and $y' \geq y$ then certainly $y' \not\leq s$, and inaccessible from below, as if D is directed with $D \cap s_{\not\leq}$ then s is an upper bound for D and so $\sup D \leq s$, so is not in $s_{\not\leq}$. Further $y \in O$ if and only if $x \leq y$: if $x \not\leq y$ then $y_{\not\leq} \supseteq O$ and so $y \notin O$. By construction $x \in O$ as $x \in s_{\not\leq}$ when $x \not\leq s$, and so $y \in O$ for $y \geq x$ as O is upwards closed.

Then let $k = \mathbb{1}_O$, the indicator function on the set O, which is continuous as O is open. Then:

$$\xi_k = \sum_i \delta(x_i, \phi_i)_k = \sum_i \phi_i \circ k(x_i) = \sum_{\substack{i \in I \\ x_i \geq x}} \phi_i$$

and similarly:

$$\xi_k = \sum_{\substack{j \in J \\ x'_j \geq x}} \phi'_j$$

However, by maximality of x we have that:

$$\sum_{\substack{i\in I\\ x_i>x}} \phi_i = \sum_{\substack{j\in J\\ x'_j\geq x}} \phi'_j$$

and so:

$$\sum_{\substack{i\in I\\ x_i=x}} \phi_i = \sum_{\substack{j\in J\\ x'_j=x}} \phi'_j$$

contradicting the construction of x.

To show $F^\dagger$ is a natural transformation we need the following to commute:

$$\begin{array}{ccc} \mathcal{Q}_{\mathrm{f}}(\mathscr{A},\mathscr{B},X) & \xrightarrow{F^\dagger_{\mathscr{A},\mathscr{B},X}} & \mathcal{Q}_{\mathrm{f}}(F(\mathscr{A}),F(\mathscr{B}),X) \\ \downarrow{\scriptstyle \mathcal{Q}_{\mathrm{f}}(\phi,\psi,f)} & & \downarrow{\scriptstyle \mathcal{Q}_{\mathrm{f}}(F(\phi),F(\psi),f)} \\ \mathcal{Q}_{\mathrm{f}}(\mathscr{A}',\mathscr{B}',Y) & \xrightarrow{F^\dagger_{\mathscr{A}',\mathscr{B}',Y}} & \mathcal{Q}_{\mathrm{f}}(F(\mathscr{A}'),F(\mathscr{B}'),Y) \end{array}$$

For $\phi \in \mathbf{W}^*(\mathscr{A},\mathscr{A}')$, $\psi \in \mathbf{W}^*(\mathscr{B}',\mathscr{B})$, $f : \mathbf{DCPO}(X,Y)$. Taking some decomposition:

$$\sum_i \delta(x_i,\chi_i) \in \mathcal{Q}_{\mathrm{f}}(\mathscr{A},\mathscr{B},X)$$

we get:

$$\begin{aligned} \mathcal{Q}_{\mathrm{f}}(F(\phi),F(\psi),f)\left(F^\dagger_{\mathscr{A},\mathscr{B},X}\left(\sum_i \delta(x_i,\chi_i)\right)\right) &= \mathcal{Q}_{\mathrm{f}}(F(\phi),F(\psi),f)\left(\sum_i \delta(x_i,F(\chi_i))\right) \\ &= \sum_i \delta(f(x_i),F(\phi)\circ F(\chi_i)\circ F(\psi)) \\ &= \sum_i \delta(f(x_i),F(\phi\circ\chi_i\circ\psi)) \\ &= F^\dagger_{\mathscr{A}',\mathscr{B}',Y}\left(\sum_i \delta(f(x_i),\phi\circ\chi_i\circ\psi)\right) \\ &= F^\dagger_{\mathscr{A}',\mathscr{B}',Y}\left(\mathcal{Q}_{\mathrm{f}}(\phi,\psi,f)\left(\sum_i \delta(x_i,\chi_i)\right)\right) \end{aligned}$$

And so $F^\dagger$ is natural. □

Hence, all the operations of the monad preserve the finiteness of quantum orchestras. Of course, the set of finite quantum orchestras on a (non-finite) dcpo is not itself a dcpo, and so $\mathcal{Q}_{\mathrm{f}}$ it not itself a monad. We can however study the order structure on this set.

Lemma 45. *If $x \leq y$ and $\Phi \leq \Psi$, then $\delta(x,\Phi) \leq \delta(y,\Psi)$. Conversely, if $\delta(x,\Phi) \leq \delta(y,\Psi)$ then $\Phi \leq \Psi$ and if $\Phi \neq 0$ then $x \leq y$.*

Proof. Suppose $x \leq y$ and $\Phi \leq \Psi$. Then for all k:

$$\delta(x,\Phi)_k = \Phi\circ k(x) \leq \Psi \circ k(y) = \delta(y,\Psi)$$

as composition of channels is monotone. Conversely assume $\delta(x,\Phi) \leq \delta(y,\Psi)$. Then let k be the continuation such that $k(z) = \mathrm{id}$ for all z. Then:

$$\Phi = \Phi\circ k(x) = \delta(x,\Phi) \leq \delta(y,\Psi) = \Psi\circ k(y) = \Psi$$

Now consider k such that:

$$k(z) = \begin{cases} \mathrm{id} & \text{if } z \not\leq y \\ 0 & \text{otherwise} \end{cases}$$

Then $\delta(y, \Psi)_k = 0$, hence $\delta(x, \Phi) = 0$ and so either $\Phi = 0$ or $x \leq y$. □

Despite the relative simplicity of the order structure on Dirac quantum orchestras, the story for finite quantum orchestras is far less trivial. We conjecture that, for finite ξ and ξ' with $\xi \leq \xi'$, one can realise this inequality as a finite sequence of inequalities from Lemma 45 on a single Dirac orchestra. A corresponding theorem is for classical valuations can be proved using the max-flow min-cut theorem (see [24]), but this theorem depends on the lattice structure of the real numbers, and does not hold for quantum channels. Despite the proof not generalising, we have found no counterexample to this statement, and it is unclear if it holds. For a detailed discussion of the link between orchestras and valuations, see Appendix D.

D Comparison to the Probabilistic Powerdomain Monad

The goal of this section is to clarify the relation of the quantum orchestra monad with the probabilistic powerdomain monad. To this end, recall that the probabilistic powerdomain monad assigns to a dcpo X the set of valuations on X, denoted $\mathcal{V}(X)$.

Definition 46 (Scalar-valued valuations [24])**.** Let X be a topological space, and $O(X)$ its complete lattice of open subsets. A function $v : O(X) \to [0, 1]$ is a *(scalar-valued) valuation* if and only if it is

- monotone, *i.e.*, $v(U) \leq v(V)$ for all $U \subseteq V$,
- strict, *i.e.*, $v(\emptyset) = 0$,
- modular, *i.e.*, $v(U) + v(V) = v(U \cup V) + v(U \cap V)$,
- continuous, *i.e.*, for any directed set $\{U_\lambda\}_{\lambda \in \Lambda}$ of open sets, it holds that

$$v\left(\bigcup_\lambda U_\lambda\right) = \sup_\lambda v\left(U_\lambda\right).$$

Theorem 47 ([24])**.** *Let X be a dcpo. Then, $\mathcal{V}(X)$ is the set of (scalar-valued) valuations on X with the Scott topology, ordered by:*

$$v \leq w \quad :\Leftrightarrow \quad v(U) \leq w(U)\ \forall U \in O(X).$$

Also, $\mathcal{V}(X)$ is a pointed dcpo with directed suprema defined pointwise and the zero valuation as its least element. Finally, $\mathcal{V}$ is a monad on **DCPO***, called the* probabilistic powerdomain *monad, with the unit $\epsilon_X : X \to \mathcal{V}(X)$ given by*

$$\epsilon_X(x)(U) = \begin{cases} 1, & \text{if } x \in U, \\ 0, & \text{otherwise.} \end{cases}$$

and the Kleisli extension $f^\sharp : \mathcal{V}(X) \to \mathcal{V}(Y)$ for continuous functions $f : X \to \mathcal{V}(Y)$ given by

$$f^\sharp(v)(V) = \int_{x \in X} f(x)(V) dv.$$

Note that the notion of valuation can be generalised almost verbatim from scalar-valued to $\mathbf{W}^*(\mathscr{B}, \mathscr{A})$-valued maps as follows.

Definition 48 ($\mathbf{W}^*(\mathscr{B}, \mathscr{A})$-valued valuations)**.** Let $\mathscr{A}, \mathscr{B} \in \mathbf{W}^*$, X be a topological space, and $O(X)$ its complete lattice of open subsets. A function $v : O(X) \to \mathbf{W}^*(\mathscr{B}, \mathscr{A})$ is called a valuation if and only if it is

- monotone, *i.e.*, $v(U) \leq v(V)$ for all $U \subseteq V$,
- strict, *i.e.*, $v(\emptyset) = 0$,
- modular, *i.e.*, $v(U) + v(V) = v(U \cup V) + v(U \cap V)$,
- continuous, *i.e.*, for any directed set $\{U_\lambda\}_{\lambda\in\Lambda}$ of open sets, it holds that

$$v\left(\bigcup_\lambda U_\lambda\right) = \sup_\lambda v\left(U_\lambda\right).$$

By identifying $\mathbf{W}^*(\mathbb{C},\mathbb{C})$ with the real interval $[0,1]$, $\mathbf{W}^*(\mathbb{C},\mathbb{C})$-valued valuations are in fact scalar-valued valuations.

D.1 Mapping orchestras to valuations

Let now $X \in \mathbf{DCPO}$. We note that any quantum orchestra $\xi \in \boldsymbol{Q}(\mathscr{A},\mathscr{B},X)$ restricts to a $\mathbf{W}^*(\mathscr{B},\mathscr{A})$-valued valuation v^ξ by taking

$$v^\xi(U) = \xi_{\mathbb{1}_U},$$

where $\mathbb{1}_U$ is the indicator function $X \to \mathbf{W}^*(\mathscr{B},\mathscr{B})$:

$$\mathbb{1}_U(x) = \begin{cases} \mathrm{id}_{\mathscr{B}}, & \text{if } x \in U, \\ 0, & \text{otherwise.} \end{cases}$$

The continuity of $\mathbb{1}_U$ corresponds directly to U being a Scott-open subset of X. By the subconvexity property of ξ, we obtain

$$v^\xi(\emptyset) = \xi_{\mathbb{1}_\emptyset} = \xi_0 = 0$$

and

$$v^\xi(U\cup V) + v^\xi(U\cap V) = \xi_{\mathbb{1}_{U\cup V}} + \xi_{\mathbb{1}_{U\cap V}} = 2\left(\frac{1}{2}\xi_{\mathbb{1}_{U\cup V}} + \frac{1}{2}\xi_{\mathbb{1}_{U\cap V}}\right) = 2\xi_{\frac{1}{2}\mathbb{1}_{U\cup V}+\frac{1}{2}\mathbb{1}_{U\cap V}} = 2\xi_{\frac{1}{2}\mathbb{1}_U+\frac{1}{2}\mathbb{1}_V}$$
$$= 2\left(\frac{1}{2}\xi_{\mathbb{1}_U} + \frac{1}{2}\xi_{\mathbb{1}_V}\right) = \xi_{\mathbb{1}_U} + \xi_{\mathbb{1}_V} = v^\xi(U) + v^\xi(V).$$

By monotonicity we immediately have for $U \subseteq V$ that

$$v^\xi(U) = \xi_{\mathbb{1}_U} \leq \xi_{\mathbb{1}_V} = v^\xi(V),$$

and by continuity we get that if $\{U_\lambda\}_{\lambda\in\Lambda}$ is a directed set of open subsets, then it also follows that

$$\sup_\lambda v^\xi(U_\lambda) = \sup_\lambda \xi_{\mathbb{1}_{U_\lambda}} = \xi_{\sup_\lambda \mathbb{1}_{U_\lambda}} = \xi_{\mathbb{1}_{\cup_\lambda U_\lambda}} = v^\xi\left(\bigcup_\lambda U_\lambda\right),$$

and so v^ξ is a valuation. The pointwise order or quantum orchestras immediately implies for orchestras $\xi,\zeta \in \boldsymbol{Q}(\mathscr{A},\mathscr{B},X)$ that

$$\xi \leq \zeta \quad\Rightarrow\quad \forall U \in O(X) : \xi_{\mathbb{1}_U} \leq \zeta_{\mathbb{1}_U} \quad\Rightarrow\quad v^\xi \leq v^\zeta,$$

and thus the mapping $\xi \mapsto v^\xi$ is a monotone mapping from orchestras to valuations.

While we have obtained a monotone mapping from orchestras to valuations in this way, it is not immediately clear whether this mapping is injective in the general case. The difficulty in showing injectivity lies in the absence of an existing non-commutative integration theory for $\mathbf{W}^*(\mathscr{B},\mathscr{A})$-valued valuations. We leave this question for future exploration, and subsequently focus on the scalar-valued case, in which a well-developed integration theory exists [24].

D.2 The scalar-valued case

Now consider the (non-parameterised) monad obtained as $\mathcal{Q}(\mathbb{C},\mathbb{C},-)$. By the arguments from Section D.1, every orchestra $\xi \in \mathcal{Q}(\mathbb{C},\mathbb{C},-)$ gives rise to a scalar-valued valuation $v^\xi : O(X) \to [0,1]$, where we identified the real interval $[0,1]$ with $\mathbf{W}^*(\mathbb{C},\mathbb{C})$. By use of the integration theory for scalar-valued valuations [24], any such valuation gives rise to a continuous integral of any continuous function $f : X \to [0,1]$:

$$f \mapsto \int f dv^\xi.$$

By the definition of v^ξ, this integral needs to coincide with the orchestra on indicator functions:

$$\int \mathbb{1}_U dv^\xi = v^\xi(U) = \xi_{\mathbb{1}_U},$$

and by linearity also on simple functions, *i.e.*, linear combinations of indicator functions. As both the integral and the orchestra are continuous, they preserve suprema, and the equality extends to all continuous functions, as all continuous functions can be represented as suprema of simple functions. We thus record: for all continuous functions $f : X \to [0,1]$, it holds that:

$$\int f dv^\xi = \xi_f.$$

We now proceed to show that in the scalar-valued case, the mapping from orchestras to valuations is injective. To this end, let $\xi, \zeta \in \mathcal{Q}(\mathbb{C},\mathbb{C},-)$ with $\xi \neq \zeta$. There then exists a W*-algebra $\mathscr{K}$ and a continuation $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K},\mathbb{C}))$ such that $\xi_k \neq \zeta_k$, where $\xi_k, \zeta_k \in \mathbf{W}^*(\mathscr{K},\mathbb{C})$. There thus exists some $m \in \mathbf{W}^*(\mathbb{C},\mathscr{K})$ such that $\xi_k \circ m \neq \zeta_k \circ m$. Define the continuous function

$$g : X \to [0,1], \quad x \mapsto k(x) \circ m.$$

Using the compositionality of orchestras, we conclude that

$$\int g dv^\xi = \xi_g = \xi_k \circ m \neq \zeta_k \circ m = \zeta_g = \int g dv^\zeta,$$

and thus $\xi^v \neq \zeta^v$. The mapping $\xi \mapsto v^\xi$ is hence injective.

Finally, let $\Delta \subseteq \mathcal{Q}(\mathbb{C},\mathbb{C},X)$ be directed. Using that scalar-valued valuations form a dcpo [24], it then follows for all $U \in O(X)$ that

$$\begin{aligned}\sup\left\{v^\xi \mid \xi \in \Delta\right\}(U) &= \sup\left\{v^\xi(U) \mid \xi \in \Delta\right\} = \sup\left\{\xi_{\mathbb{1}_U} \mid \xi \in \Delta\right\} \\ &= \left[\sup\{\xi \mid \xi \in \Delta\}\right]_{\mathbb{1}_U} = v^{[\sup\{\xi \mid \xi \in \Delta\}]}(U).\end{aligned}$$

The mapping $\xi \mapsto v^\xi$ is thus also preserving suprema and hence continuous. We summarise these results in the following proposition.

Proposition 49. *For any $X \in \mathbf{DCPO}$, there exists an injective, continuous map $\alpha_X : \mathcal{Q}(\mathbb{C},\mathbb{C},X) \to \mathcal{V}(X)$, $\xi \mapsto v^\xi$.*

D.3 Mapping scalar-valued valuations to orchestras

Having established that there is an injective, continuous mapping from $\mathcal{Q}(\mathbb{C},\mathbb{C},X)$ to the dcpo of valuations $O(X) \to [0,1]$, we now ask the question whether this mapping is invertible.

To this end, let $v : O(X) \to [0,1]$ be a valuation, let $\mathscr{K} \in \mathbf{W}^*$, and consider a continuation $k \in \mathbf{DCPO}(X, \mathbf{W}^*(\mathscr{K},\mathbb{C}))$. As a first step towards constructing an integral of k along v, define

$$\xi_k^v : \mathscr{K}^+ \to \mathbb{R}^+, \quad \kappa \mapsto \int k(\kappa) dv,$$

using that $k(\kappa)$ is always a positive real number for all $\kappa \in \mathscr{K}^+$. The positive cone $\mathscr{K}^+$ linearly spans the full algebra $\mathscr{K}$ as every element can be decomposed into a linear combination of four positive elements. Linearly extending to all of $\mathscr{K}$, we obtain a map $\xi_k^v : \mathscr{K} \to \mathbb{C}$. The positivity of ξ_k^v is immediate, complete positivity is equivalent to positivity for functionals, and normality of ξ_k^v follows from the fact that normality on W*-algebras is equivalent to the preservation of suprema [9] and the fact that both k and integration along valuations are Scott-continuous. We thus showed that $\xi_k^v \in \mathbf{W^*}(\mathscr{K}, \mathbb{C})$. As a candidate orchestra, we have obtained $\xi^v = \{\xi_k^v \in \mathbf{W^*}(\mathscr{K}, \mathbb{C}) \mid \mathscr{K} \in \mathbf{W^*}, k \in \mathbf{DCPO}(X, \mathbf{W^*}(\mathscr{K}, \mathbb{C}))\}$.

It remains to show that ξ^v is continuous as a mapping $k \mapsto \xi_k^v$, subconvex, and satisfies compositionality. Subconvexity and compositionality are direct consequences of the linear construction of the candidate orchestra. To prove continuity, let $K \subseteq \mathbf{DCPO}(X, \mathbf{W^*}(\mathscr{K}, \mathbb{C}))$ be directed. It then follows for $\kappa \in \mathscr{K}^+$ that

$$\begin{aligned}\xi^v_{\sup K}(\kappa) &= \int [\sup K](\kappa) dv = \int \sup\{k(\kappa) \mid k \in K\}\, dv = \sup\left\{ \int k(\kappa) dv \,\middle|\, k \in K \right\} \\ &= \sup\left\{ \xi_k^v(\kappa) \,\middle|\, k \in K \right\} = \left[\sup\left\{ \xi_k^v \,\middle|\, k \in K \right\}\right](\kappa),\end{aligned}$$

and thus by linearity that $\xi^v_{\sup K} = \sup\left\{\xi_k^v \mid k \in K\right\}$. We conclude that ξ^v is a quantum orchestra. We have thus constructed a map $\beta_X : \mathcal{V}(X) \to \mathcal{Q}(\mathbb{C}, \mathbb{C}, X)$, $v \mapsto \xi^v$.

By construction of β_X, it holds that $v = \alpha_X[\beta_X[v]]$ for all $v \in \mathcal{V}(X)$. This implies that α_X from Section D.2 must have been surjective, showing that it is in fact bijective, and β_X its inverse.

For the final goal to establish an isomorphism between scalar-valued orchestras and valuations on **DCPO** it remains to show that this inverse is Scott-continuous.

In order to show monotonicity of the inverse, start with two comparable valuations $v, w : O(X) \to [0, 1]$ with $v \leq w$. Let ξ^v, ξ^w be the two arising orchestras constructed as above. For any $\mathscr{K} \in \mathbf{W^*}$, any continuation $k \in \mathbf{DCPO}(X, \mathbf{W^*}(\mathscr{K}, \mathbb{C}))$, and $\kappa \in \mathscr{K}^+$, it holds that

$$\xi_k^v(\kappa) = \int k(\kappa) dv \leq \int k(\kappa) dw = \xi_k^w(\kappa),$$

and thus $\xi_k^v \leq \xi_k^w$ in the Löwner order for all k, implying that $\xi^v \leq \xi^w$. The map β_X is thus monotone.

To prove the preservation of suprema, let $K \subseteq \mathcal{V}(X)$ be a directed set of valuations. It then holds for any $\mathscr{K} \in \mathbf{W^*}$, any continuation $k \in \mathbf{DCPO}(X, \mathbf{W^*}(\mathscr{K}, \mathbb{C}))$, and for any $\kappa \in \mathscr{K}^+$ that

$$\begin{aligned}\left[\xi^{\sup K}\right]_k(\kappa) &= \int k(\kappa) d[\sup K] = \sup\left\{ \int k(\kappa) dv \,\middle|\, v \in K \right\} \\ &= \sup\left\{ \xi_k^v(\kappa) \,\middle|\, v \in K \right\} = \left[\sup\left\{ \xi_k^v \,\middle|\, v \in K \right\}\right](\kappa) = \left[\sup\{\, \xi^v \mid v \in K\}\right]_k(\kappa),\end{aligned}$$

using the continuity of the integral in the valuation. It follows that $\xi^{\sup K} = [\sup\{\, \xi^v \mid v \in K\}]$, and thus β_X is continuous. This fact implies the next proposition.

Proposition 50. *For any $X \in \mathbf{DCPO}$, α_X is a $\mathbf{DCPO}$-isomorphism $\mathcal{Q}(\mathbb{C}, \mathbb{C}, X) \to \mathcal{V}(X)$.*

D.4 Monad equivalence

We now aim to prove that $\mathcal{Q}(\mathbb{C}, \mathbb{C}, -)$ and $\mathcal{V}$ are equivalent. We firstly show that α_X forms in fact a natural isomorphism between the functors $\mathcal{Q}(\mathbb{C}, \mathbb{C}, -)$ and $\mathcal{V}$. To this end, we must show that the

following diagram commutes for all morphisms $f \in \mathbf{DCPO}(X, Y)$.

$$\begin{array}{ccc} \mathcal{Q}(\mathbb{C},\mathbb{C},X) & \xrightarrow{\alpha_X} & \mathcal{V}(X) \\ {\scriptstyle \mathcal{Q}(\mathbb{C},\mathbb{C},f)}\downarrow & & \downarrow{\scriptstyle \mathcal{V}(f)} \\ \mathcal{Q}(\mathbb{C},\mathbb{C},Y) & \xrightarrow{\alpha_Y} & \mathcal{V}(Y) \end{array}$$

Let now $\xi \in \mathcal{Q}(\mathbb{C},\mathbb{C},X)$ and $V \in O(Y)$. Then, by definition, it holds that

$$\begin{aligned} [\alpha_Y \circ \mathcal{Q}(\mathbb{C},\mathbb{C},f)(\xi)]\,(V) &= \mathcal{Q}(\mathbb{C},\mathbb{C},f)(\xi)_{\mathbb{1}_V} = \xi_{\lambda x.\mathbb{1}_V(f(x))} = \xi_{\mathbb{1}_{f^{-1}(V)}} \\ &= \alpha_X(\xi)(f^{-1}(V)) = [\mathcal{V}(f) \circ \alpha_X(\xi)]\,(V), \end{aligned}$$

which shows that the diagram indeed commutes, proving the naturality of α_X and the following claim.

Proposition 51. *$\mathcal{Q}(\mathbb{C},\mathbb{C},-)$ and $\mathcal{V}$ are functorially equivalent on* **DCPO**.

We proceed with proving the compatibility of the natural transformation α_X with the units of the two monads. To this end, we need to show that the diagram

$$\begin{array}{ccc} X & \xrightarrow{\eta^{\mathbb{C}}X} & \mathcal{Q}(\mathbb{C},\mathbb{C},X) \\ & {\scriptstyle \epsilon_X}\searrow & \downarrow{\scriptstyle \alpha_X} \\ & & \mathcal{V}(X) \end{array}$$

commutes. Let $x \in X$ for some $X \in \mathbf{DCPO}$, and $U \in O(X)$. Then it holds that

$$[\alpha_X \circ \eta_{\mathbb{C}X}(x)]\,(U) = [\eta_{\mathbb{C}X}(x)]_{\mathbb{1}_U} = \mathbb{1}_U(x) = \epsilon_X(x)(U),$$

and thus α_X is compatible with the units.

To further show that α_X is compatible with the multiplications of the two monads, we need to prove the commutativity of the following diagram:

$$\begin{array}{ccccccc} & & & & \mathcal{Q}(\mathbb{C},\mathbb{C},\mathcal{V}(X)) & & \\ & & {\scriptstyle \mathcal{Q}(\mathbb{C},\mathbb{C},\alpha_X)}\nearrow & & & \searrow{\scriptstyle \alpha_{\mathcal{V}(X)}} & \\ \mathcal{Q}(\mathbb{C},\mathbb{C},\mathcal{Q}(\mathbb{C},\mathbb{C},X)) & \xrightarrow[\mu_x]{} & \mathcal{Q}(\mathbb{C},\mathbb{C},X) & \xrightarrow{\alpha_X} & \mathcal{V}(X) & \xleftarrow{\nu_X} & \mathcal{V}(\mathcal{V}(X)) \\ & & {\scriptstyle \alpha_{\mathcal{Q}(\mathbb{C},\mathbb{C},X)}}\searrow & & & \nearrow{\scriptstyle \mathcal{V}(\alpha_X)} & \\ & & & & \mathcal{V}(\mathcal{Q}(\mathbb{C},\mathbb{C},X)) & & \end{array}$$

To this end, let $\chi \in \mathcal{Q}(\mathbb{C},\mathbb{C},\mathcal{Q}(\mathbb{C},\mathbb{C},X))$ and $U \in O(X)$. It then holds

$$[\alpha_X \circ \mu_X(\chi)]\,(U) = [\mu_X(\chi)]_{\mathbb{1}_U} = \chi_{\lambda\xi.\xi_{\mathbb{1}_U}},$$

but also

$$\begin{aligned} \left[\nu_X \circ \alpha_{\mathcal{V}(X)} \circ \mathcal{Q}(\mathbb{C},\mathbb{C},\alpha_X)(\chi)\right](U) &= \int_{\mu \in \mathcal{V}(X)} \mu(U)\, d\left(\alpha_{\mathcal{V}(X)} \circ \mathcal{Q}(\mathbb{C},\mathbb{C},\alpha_X)(\chi)\right) \\ &= [\mathcal{Q}(\mathbb{C},\mathbb{C},\alpha_X)(\chi)]_{\lambda\mu.\mu(U)} = \chi_{\lambda\xi.\xi[\lambda\mu.\mu(U)](\alpha_X(\xi))} \\ &= \chi_{\lambda\xi.[\alpha_X(\xi)](U)} = \chi_{\lambda\xi.\xi_{\mathbb{1}_U}}. \end{aligned}$$

The equality of these two expressions, together with the naturality of α_X, shows that above diagram commutes, and that the natural isomorphism α_X is compatible with the multiplication. All in all, we arrive at the main result of this section.

Theorem 52. *The monad* $\mathcal{Q}(\mathbb{C}, \mathbb{C}, -)$ *is isomorphic to the probabilistic powerdomain monad* $\mathcal{V}$.