Free Quantum Computing

Robin Kaarsgaard University of Southern Denmark

Jacques Carette McMaster University **Chris Heunen** University of Edinburgh **Neil Julien Ross** Dalhousie University **Amr Sabry** Indiana University

Two ideas

(by people much smarter than me)

Formalising useful abstractions in quantum computing

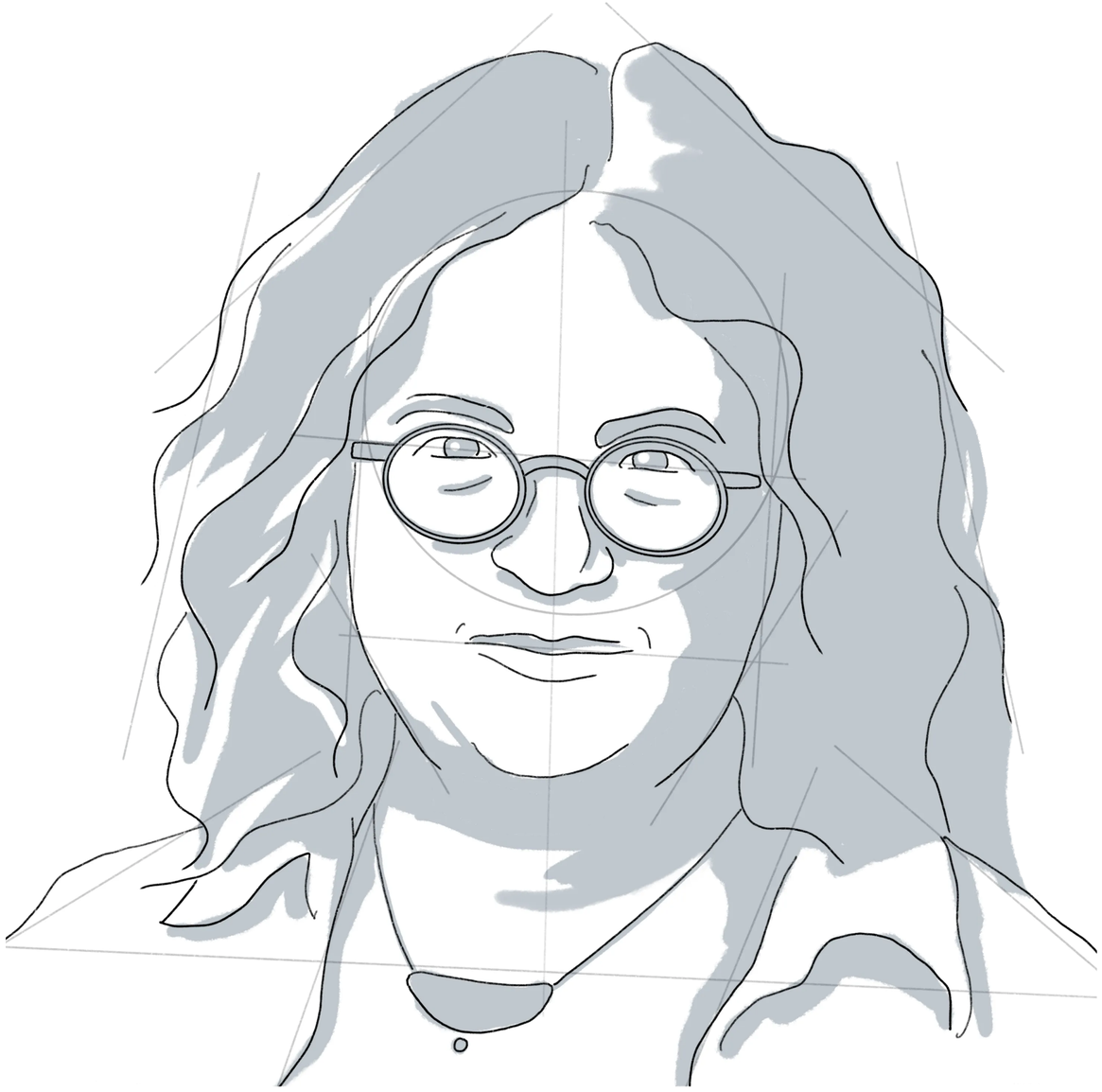
"[F]ormalizing useful abstractions in quantum computing for the manipulation of quantum information [..] is still nascent, and may have great use in the development of higher level quantum programming languages as we attempt to move away from a gate-level understanding of quantum computations."

Formalising useful abstractions in quantum computing

- Developing a high-level understanding of quantum computing, for example to lead the development of quantum programming languages
- Developing abstractions for quantum computing, for example to allow us to formalise results about quantum computing in a proof assistant, in a way that mirrors our high-level understanding.

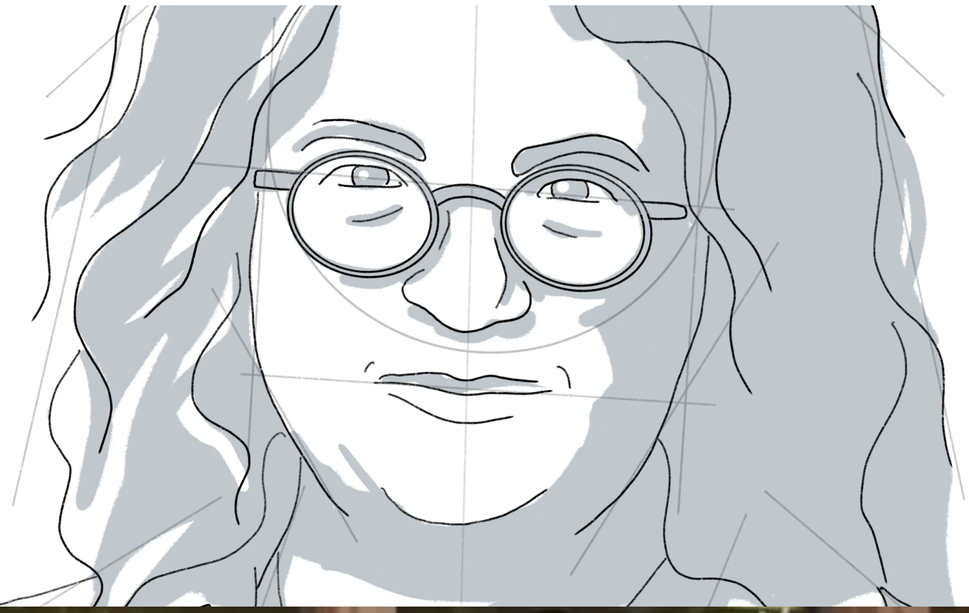
Quantum computation as a completion

“It’s really something that is special for quantum computation because it’s somehow ‘complete’ — quantum computation is some kind of completion, mathematically, of classical computation. I think of this as maybe similar to the fact that the complex numbers are an algebraic closure of the real numbers.”



Aharonov-Shi theorem

- The Toffoli gate is universal for classical reversible computing.
- The Hadamard gate is fundamentally quantum: it introduces and eliminates uniform superpositions of states.
- **Theorem (Aharonov-Shi):** Toffoli+Hadamard is universal for quantum computing.



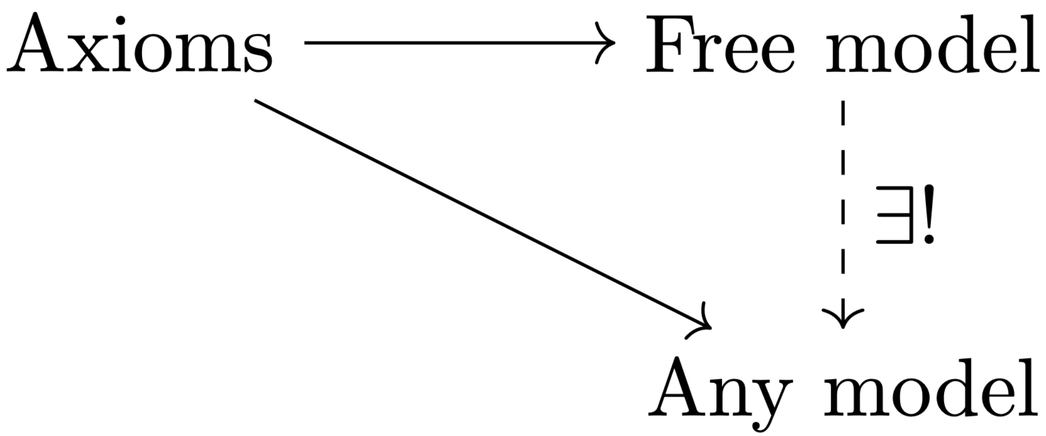
In this talk

- I’ll describe a framework which formalises quantum computation into a small set of generating maps and equations which
  - introduces the ability to stop (certain) classical computations partway as a fundamental abstraction of quantum computation,
  - exhibits quantum computing as a completion of classical (reversible) computing,
  - can faithfully replace the linear algebraic formulation of quantum computation, and is
  - implementable directly in modern proof assistants, such as Agda or Lean.

Free quantum computing?

Free models

- The defining property of a free model:
  - it realises our axioms of quantum computation, and there is a unique interpretation of the free model in any other model satisfying our axioms.
  - In this sense, the free model contains exactly what is needed to form a model, and nothing more.



Starting point: Rig categories

$$\frac{f : X \to Y \quad g : Y \to Z}{g \circ f : X \to Z}$$

$$\frac{}{\mathrm{id}_X : X \to X}$$

$$(f \circ g) \circ h = f \circ (g \circ h)$$
$$\mathrm{id}_Y \circ f = f = f \circ \mathrm{id}_X$$

$$\frac{f : X \to Y \quad g : X' \to Y'}{f \otimes g : X \otimes X' \to Y \otimes Y'}$$

$$(X \otimes Y) \otimes Z \simeq X \otimes (Y \otimes Z)$$
$$X \otimes I \simeq X \simeq I \otimes X$$
$$X \otimes Y \simeq Y \otimes X$$

$$\frac{f : X \to Y \quad g : X' \to Y'}{f \oplus g : X \oplus X' \to Y \oplus Y'}$$

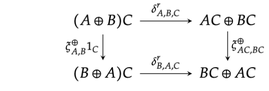
$$(X \oplus Y) \oplus Z \simeq X \oplus (Y \oplus Z)$$
$$X \oplus O \simeq X \simeq O \oplus X$$
$$X \oplus Y \simeq Y \oplus X$$

$$X \otimes (Y \oplus Z) \simeq (X \otimes Y) \oplus (X \otimes Z)$$
$$X \otimes O \simeq O$$

Rig categories and finite bijections

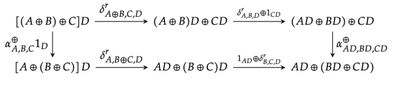
- Rig categories can express all generators of symmetric groups: it is *universal* for finite bijections (i.e., it can express all of them).
- Rig categories with only invertible maps are called *rig groupoids*.
- Rig categories must satisfy a number of equations, *coherence conditions*.
- These equations are complete (*fully abstract*) for finite bijections, in the sense that any two finite bijections π_1 and π_2 are equal (as functions) if and only if it can be shown using the coherence conditions.

(2.1.7)

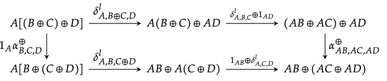


Distributivity and Additive Associativity:

(2.1.8)

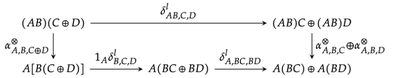


(2.1.9)

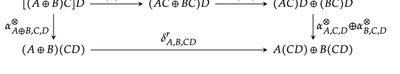


Distributivity and Multiplicative Associativity:

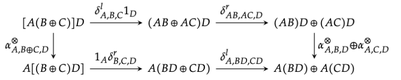
(2.1.10)



(2.1.11)

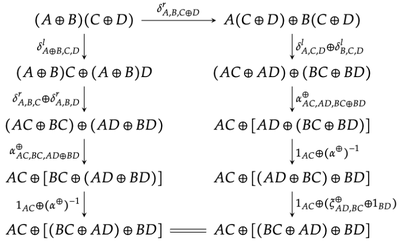


(2.1.12)



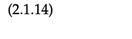
2-By-2 Distributivity:

(2.1.13)



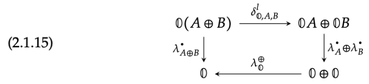
Multiplicative Zero of 0:

(2.1.14)

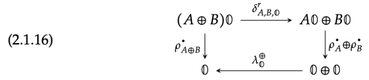


Multiplicative Zero of a Sum:

(2.1.15)

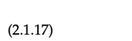


(2.1.16)

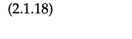


Multiplicative Zero and Multiplicative Unit:

(2.1.17)



(2.1.18)

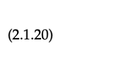


Symmetry of Multiplicative Zero:

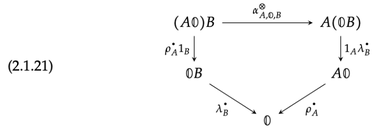
(2.1.19)

Multiplicative Zero and Multiplicative Associativity:

(2.1.20)



(2.1.21)



Sprinkle in a few square roots: Π_k

We extend the language of rig groupoids by adding two base isomorphisms

$$\zeta_k : I \leftrightarrow I \qquad \mathsf{V} : I \oplus I \leftrightarrow I \oplus I$$

and four axioms governing them

(E1) $\mathsf{V}^2 = \mathsf{X} = \sigma^{\oplus}_{I \oplus I}$

(E2) $\zeta_k^{2^k} = \mathrm{id}_1$

(E3) $\mathsf{V} \circ \mathsf{S} \circ \mathsf{V} = \mathsf{S} \circ \mathsf{V} \circ \mathsf{S}$

(E4) if $f \oplus g = f' \oplus g'$ then $f = f'$ (and $g = g'$)

where $\mathsf{S} = \mathrm{id} \oplus \zeta_k^{2^{k-2}}$ and $k \geq 2$. We call the resulting language Π_k (first introduced under the name $\sqrt{\Pi} = \Pi_3$)

Models of Π_k

- Models of Π_k are rig groupoids $(\mathbf{C}, I, O, \otimes, \oplus)$ with distinguished maps $\zeta_k : I \to I$ and $\mathsf{V} : I \oplus I \to I \oplus I$ such that
  - There is a mapping $[\![\cdot]\!]$ of types to objects and programs to morphisms satisfying $[\![\zeta_k]\!] = \zeta_k$, $[\![\mathsf{V}]\!] = \mathsf{V}$, and the axioms of a *rig functor.*
  - $f = g$ implies $[\![f]\!] = [\![g]\!]$.

Models of Π_k

- The category **Unitary** of finite dimensional Hilbert spaces and unitaries is a rig category: tensor product as $\otimes$, and direct sum as $\oplus$.
- Since every unitary is invertible, this is a rig groupoid.
- Choosing

$$\zeta_k = e^{2\pi i/2^k} \qquad \mathsf{V} = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

we see that **Unitary** is a model of Π_k.

Models of Π_k

- Unlike **Unitary,** Π_k is presented by a discrete set of generators and relations.
  - It corresponds more directly to programs executable on realistic quantum hardware (where only a discrete gate set is available).
  - It corresponds to a stand-alone equational system that can be directly implemented in a proof assistant.
- Π_k is computationally universal for all $k \geq 2$**.**
  - **Key question 1:** How do we make sense of the axioms?
  - **Key question 2:** What can be derived using these axioms?

Hamiltonian evolution and square roots

- The Schrödinger equation governs the evolution of a physical system described by a *Hamiltonian* H (letting some time t pass),

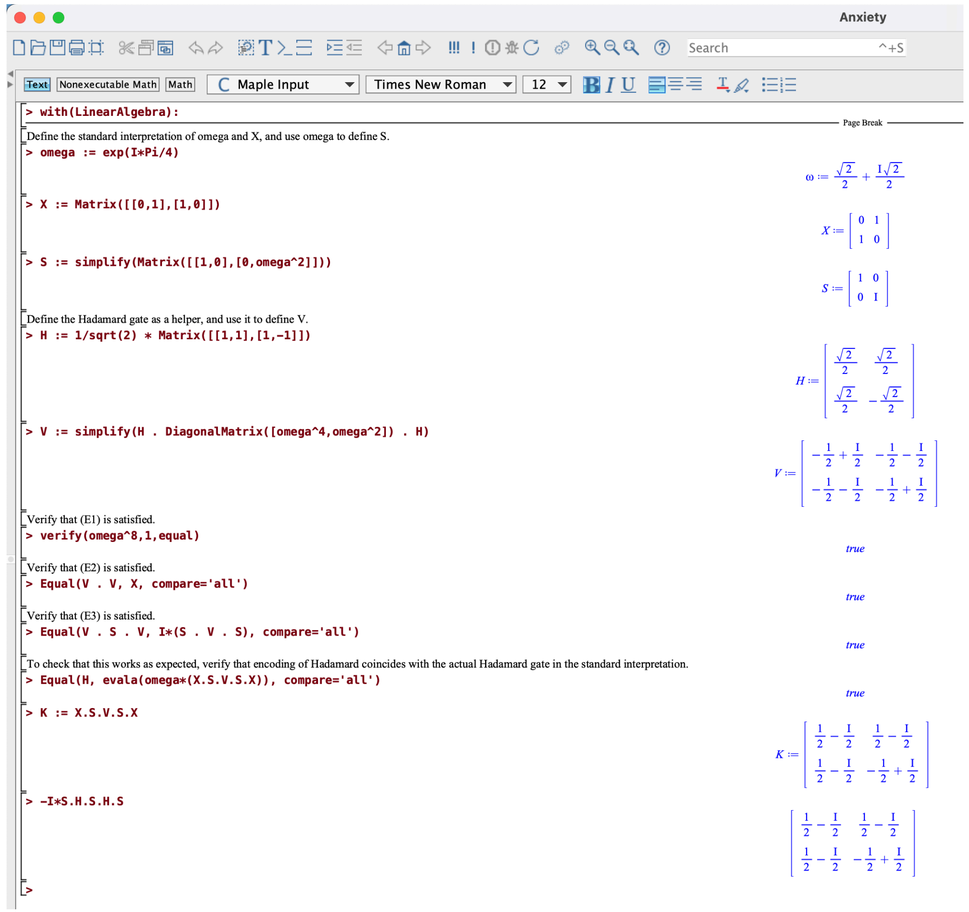
$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = H |\Psi(t)\rangle$$

- Its solutions are famously unitaries depending on t, $U(t) = e^{-itH}$.
- What happens if instead of letting time t pass, we let half as much time $\frac{t}{2}$ pass?
- We get $U(t/2) = e^{-itH/2}$, and we have

$$U(t/2)U(t/2) = e^{-itH/2}e^{-itH/2} \overset{(*)}{=} e^{-itH/2-itH/2} = e^{-itH} = U(t)$$

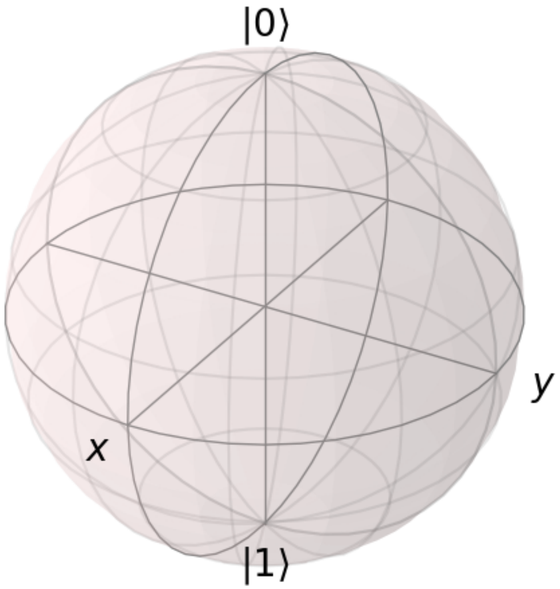
- So $U(t/2)$ acts as a (generalised) square root of $U(t)$. This motivates the first two axioms.

Qualms of the Working Scientist: Anxiety.mw



Euler decomposition

- **Fact:** Any rotation of the sphere can be decomposed into three rotations along (at least) two orthogonal axes. (XZX, ZXZ, XYZ, etc.)
  - Common system in aviation is *pitch, yaw,* and *roll*.
- Via the Bloch sphere, a choice of orthogonal axes gives a unique* decomposition of 1-qubit gates into rotation gates.
- Being able to resolve *arbitrary* Euler decompositions is necessary in the continuous case…
  - …but if we're willing to give this up in favour of a discrete-but-arbitrarily-precise case, we can get away with just one!



Euler decomposition

- Looking at the Bloch sphere, we notice that (the unitary semantics of) S and V are (respectively) 90 degree rotations along the Z and X axes.
- So the equation $S \circ V \circ S = V \circ S \circ V$ seems to relate a ZXZ Euler decomposition to an XZX one – but for which gate?
- Up to a global phase (which doesn't exist anyway), $S \circ V \circ S = V \circ S \circ V = H$, the Hadamard gate.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Graph States and the Necessity of Euler Decomposition

Ross Duncan[1] and Simon Perdrix[2,3]

[1] Oxford University Computing Laboratory,
Wolfson Building, Parks Road, OX1 3QD Oxford, UK
ross.duncan@comlab.ox.ac.uk
[2] LFCS, University of Edinburgh, UK
[3] PPS, Université Paris Diderot, France
simon.perdrix@pps.jussieu.fr

Abstract. Coecke and Duncan recently introduced a categorical formalisation of the interaction of complementary quantum observables. In this paper we use their diagrammatic language to study graph states, a computationally interesting class of quantum states. We give a graphical proof of the fixpoint property of graph states. We then introduce a new equation, for the Euler decomposition of the Hadamard gate, and demonstrate that Van den Nest's theorem—locally equivalent graphs represent the same entanglement—is equivalent to this new axiom. Finally we prove that the Euler decomposition equation is not derivable from the existing axioms.

Keywords: quantum computation, monoidal categories, graphical calculi.

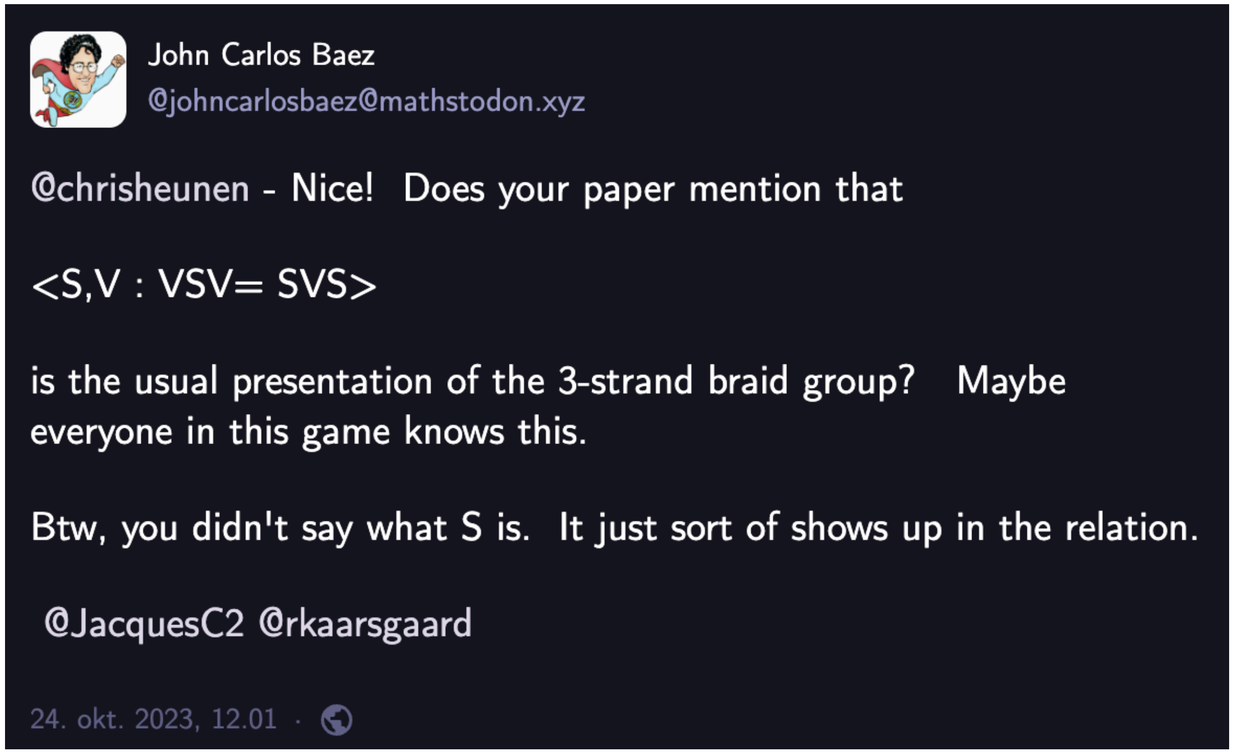
1 Introduction

Upon asking the question "What are the axioms of quantum mechanics?" we can expect to hear the usual story about states being vectors of some Hilbert space, evolution in time being determined by unitary transformations, etc. However, even before finishing chapter one of the textbook, we surely notice that something is amiss. Issues around normalisation, global phases, etc. point to an "impedence mismatch" between the theory of quantum mechanics and the mathematics used to formalise it. The question therefore should be "What are the axioms of quantum mechanics *without Hilbert spaces'?*"

In their seminal paper [1] Abramsky and Coecke approached this question by studying the categorical structures necessary to carry out certain quantum information processing tasks. The categorical treatment provides as an intuitive pictorial formalism where quantum states and processes are represented as certain diagrams, and equations between them are described by rewriting diagrams. A recent contribution to this programme was Coecke and Duncan's axiomatisation of the algebra of a pair complementary observables [2] in terms of the

K. Ambos-Spies, B. Löwe, and W. Merkle (Eds.): CiE 2009, LNCS 5635, pp. 167–177, 2009.

Drama! Enter B_3, a mysterious stranger!



John Carlos Baez
@johncarlosbaez@mathstodon.xyz

@chrisheunen - Nice! Does your paper mention that

<S,V : VSV= SVS>

is the usual presentation of the 3-strand braid group? Maybe everyone in this game knows this.

Btw, you didn't say what S is. It just sort of shows up in the relation.

@JacquesC2 @rkaarsgaard

24. okt. 2023, 12.01 ·

Unitary representations of braid groups

- So, it seems models of Π_k have some roots of distinguished maps, and contain a representation of the three-strand braid group in $\mathrm{Hom}(I \oplus I, I \oplus I)$ (??!)
- **Interesting but apparently useless fact:** The n-strand braid group B_n is isomorphic to the mapping class group of a disk with n punctures (or marked points).
- If only there was some theory concerning the movement of particles in space and relating this to quantum computation…

Topological quantum computing

- Topological quantum computing is just that! It performs computations by braiding certain quasiparticles – (non-abelian) anyons – on a two-dimensional surface.
- Performing a “native gate” corresponds to braiding two particles.
- With the usual unitary semantics for S and V,

$$\mathsf{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \qquad \mathsf{V} = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}$$

  we see that this is actually a known topological quantum computer – the Ising topological quantum computer.

- The Ising TQC is not universal, however, so Π_k can be seen as an extension with non-topological gates (e.g., via magic state distillation).

Strongly cancellative rig categories

- The final piece of the puzzle concerns the inference rule which, formulated in the language of monoidal categories, is the property that if $f \oplus g = f' \oplus g'$ implies $f = f'$ (and $g = g'$).
- This is not implied by the axioms of symmetric monoidal categories: some symmetric monoidal categories have it, others do not.
  - **Example:** $\mathbf{Unitary}$ with $\oplus$ has it.
  - **Counterexample:** $\mathbf{Unitary}$ with $\otimes$ does not.
- However, we can form a congruence that lets us make any symmetric monoidal category have this property by quotienting by it.

Models of Π_k

- Models of Π_k are rig groupoids (i.e., models of finite reversible computing) which are…
  - *Time-divisible:* Contains appropriate roots of unity and X.
  - *Topological:* The three strand braid group is canonically represented.
  - *Strongly cancellative:* The additive monoidal structure is strongly cancellative.
- In itself, Π_k is (rig equivalent to) the free strongly cancellative rig category with generators $\zeta_k : I \to I$ and $\mathsf{V} : I \oplus I \to I \oplus I$ subject to the three axioms.

Full abstraction for unitaries

- **Theorem:** For all $k \geq 2$, the rig functor $[\![\cdot]\!] : \Pi_k \to \mathbf{Unitary}$ is faithful.
- **In other words:** the axioms of Π_k are exactly the ones you need to capture the equalities of the unitaries it can express.
- **In other words:** Π_k can show *the exact same equational results* about its programs as can be shown with linear algebra, but without all of the heavy-to-implement linear algebra machinery.
- In general, Π_k is equivalent (as a rig category) to the category of natural numbers and unitary matrices generated by

$$\zeta_k = e^{2\pi i/2^k} \qquad \mathsf{V} = \frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}$$

Full abstraction for unitaries (number theory edition)

- Important special cases:
  - $\Pi_2 \simeq \mathbf{Unitary}\left(\mathbb{Z}\left[\frac{1}{2}, i\right]\right)$
  - $\Pi_3 \simeq \mathbf{Unitary}\left(\mathbb{Z}\left[\frac{1}{2}, \omega\right]\right)$ (where $\omega = \sqrt{i}$)
  - So surely, $\Pi_4 \simeq \mathbf{Unitary}\left(\mathbb{Z}\left[\frac{1}{2}, \sqrt{\omega}\right]\right)$, right?
    - No! Doesn't continue (blame number theory).

Measurement

- **Born rule:** Measuring a state $|\psi\rangle$ with an observable $P = \sum \lambda_m P_m$ results in outcome k with probability $p(k) = \langle\psi| P_k |\psi\rangle$ and post-measurement state $\frac{1}{\sqrt{p(k)}} P_k |\psi\rangle$.
- **Problem:** Measurement perturbs a system in a way which is not a consequence of the Schrödinger equation (i.e., not a unitary evolution).
  - How do we handle measurement in Π_k?

Measurement via dilations

- **Brilliant idea (Huot-Staton):** Use two successive dilations to simulate quantum channels using unitaries alone.
  - The *Gram-Schmidt procedure* allows us to complete any isometry $\mathbb{C}^n \to \mathbb{C}^m$ to a unitary $\mathbb{C}^m \to \mathbb{C}^m$ – the required auxiliary vectors are unique up to unitary.
  - *Stinespring dilation* allows us to simulate any quantum channel $M_n(\mathbb{C}) \to M_k(\mathbb{C})$ using an isometry $\mathbb{C}^n \to \mathbb{C}^m$ and a partial trace.
- Further, each of these are governed by simple universal properties relating to the two symmetric monoidal structures of **Unitary**!

Quantum channels as a categorical completion

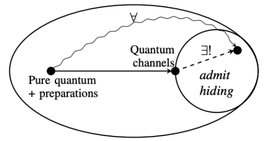
Mathieu Huot
University of Oxford, UK

Sam Staton
University of Oxford, UK

***Abstract*—We propose a categorical foundation for the connection between pure and mixed states in quantum information and quantum computation. The foundation is based on distributive monoidal categories.**

First, we prove that the category of all quantum channels is a canonical completion of the category of pure quantum operations (with ancilla preparations). More precisely, we prove that the category of completely positive trace-preserving maps between finite-dimensional C*-algebras is a canonical completion of the category of finite-dimensional vector spaces and isometries.

Second, we extend our result to give a foundation to the topological relationships between quantum channels. We do this by generalizing our categorical foundation to the topologically-enriched setting. In particular, we show that the operator norm topology on quantum channels is the canonical topology induced by the norm topology on isometries.



I. Introduction

A popular explanation of quantum theory says that, in reality, everything is reversible ("pure quantum"), but conceptually we can hide and prepare things, and this is what leads to classical data, randomness and perceived irreversibility ("full quantum"). In this paper we explain the passage from theories of pure quantum to theories of full quantum in terms of categorical completions.

We test this passage in several ways:

- Starting from pure quantum with preparations (isometries), we recover quantum channels (completely positive maps between C*-algebras) as a completion with hiding — this is our main result (Thm. V.6);
- Starting from pure quantum (unitaries), we recover preparation of ancillas (isometries) as a completion with preparations (Thm. III.3);
- Also starting from pure quantum (unitaries), we recover finite non-commutative geometry (finite-dimensional C*-algebras and $*$-homomorphisms) as a different completion (Thm. IV.10);
- Starting from topologies on the isometries, we recover topologies on quantum channels as a completion (Thm. VI.8).

All these require slightly different kinds of completion, and in this introduction we discuss the kinds of categories and completion at hand. First we consider the pure situation (§I-A), then preparation of states (§I-B), and finally hiding of states (§I-C) and topology (§I-D). In what follows we use categorical terminology, but the casual reader may prefer the following informal picture of our main result.

Informally, the outer ellipse contains all the possible theories, including pure quantum theory with preparations. The inner circle contains the theories that admit hiding. Our main result is that of all the theories that admit hiding, quantum channels are the 'closest' to pure quantum with preparations. This notion of 'closeness' will be made precise using category theory.

In [21] we presented a similar paradigm for the restricted version of quantum channels between matrix algebras. We proved that those quantum channels are the affine completion of the category of isometries, both seen as monoidal categories. We go further here by considering all finite dimensional C*-algebras which amounts to handling classical data.

A. Rudiments of pure / reversible computing

Before moving to categorical side, we recall some rudiments of reversible computing, which is one perspective on pure quantum theory. The basic idea is that a classical reversible operation on an n-level system is a bijection $n \to n$ on the natural number n considered as a finite set. A *quantum* reversible operation is an $n \times n$ complex matrix that is unitary. But the reader unfamiliar with quantum theory can focus on the classical setting for now, because every bijection can be thought of as a unitary matrix valued in $\{0, 1\}$. For example, there are two reversible classical operations on bits $2 \to 2$, identity and negation, and a reversible 2-bit operation is a bijection $4 \to 4$. The natural numbers form a rig (aka semiring) under addition and multiplication, and we find a simple calculus for building reversible operations by noticing that the bijections and unitaries can be composed but also they can be combined according to these rig operations. Here we write $(\oplus, N)$ and $(\otimes, I)$ instead of $+$ and $\times$ to emphasise their categorical nature.

- The multiplication of numbers corresponds to spatial juxtaposition of systems. For example, given two bijections on a bit, $f, g : 2 \to 2$, we have a bijection

Quantum information effects

- To add support for measurement, we add two capabilities in succession to Π_k, resulting in a new model $L(R(\Pi_k))$:
  - Add allocation – derive classical cloning.
  - Add hiding – derive measurement.
- Both of these are computational effects in that they correspond to *arrows* in the sense of Hughes.

Staton's axioms

- Staton gave a sound and complete set of equations for determining equality of quantum programs with measurement (*quantum channels*).
  - Assumed that equality of unitaries could be determined by some other means, focusing just on the semantics of measurement.
- We show that these rules can all be derived from the definition of measurement in $L(R(-))$.

Algebraic Effects, Linearity, and Quantum Programming Languages

Sam Staton
Radboud University Nijmegen

Abstract

We develop a new framework of algebraic theories with linear parameters, and use it to analyze the equational reasoning principles of quantum computing and quantum programming languages. We use the framework as follows:

- we present a new elementary algebraic theory of quantum computation, built from unitary gates and measurement;
- we provide a completeness theorem for the elementary algebraic theory by relating it with a model from operator algebra;
- we extract an equational theory for a quantum programming language from the algebraic theory;
- we compare quantum computation with other local notions of computation by investigating variations on the algebraic theory.

1. Introduction

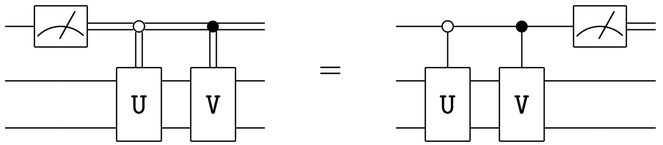
Quantum programming languages test many of the challenges of modern programming language theory: linear use of resources, separation, locality. A good way to understand a programming language is to understand equality of programs. In this paper we develop a general algebraic framework for computational effects involving linear resources. We use it to give a complete axiomatization of equality of quantum programs.

What is quantum computing? From a programming language perspective, quantum computing involves qubits and entanglement:

- *There is a type* qubit *of qubits.* Viewed as an abstract type, we can imagine a qubit as having an internal state that is a position on the surface of a sphere (called the Bloch sphere), but the accessor functions do not actually permit us to read its position on the surface. The three accessor functions are, informally, as follows. (*Notation:* we underline them.)
  - measure: make a random boolean choice, with the probability of returning either 0 or 1 depending on the Z co-ordinate of the qubit (this is called the standard basis). For example, if the qubit was on the X axis, the result of measuring will be 0 or 1 with equal probability, like tossing a fair coin; if it was at the very top of the sphere, the result of measuring will be 0 with certainty; if it was at the very bottom of the sphere, the result of measuring will be 1 with certainty. Measuring a qubit destroys it: all that remains is the result of the measurement.
  - $\mathsf{apply}_{\mathsf{X}}$: make the unitary rotation corresponding to rotating it by 180° around the X axis, taking the top of the sphere to the bottom; this unitary rotation is notated $\mathtt{X}$, and so the function that applies the rotation is notated $\mathsf{apply}_{\mathtt{X}}$.
- *For types A and B, there is a type $A \otimes B$ of entangled pairs.* For instance the type qubit $\otimes$ qubit is a type of pairs of possibly entangled qubits. Entanglement is achieved by controlled unitary rotations. For example, the controlled-$\mathtt{X}$ unitary, $\mathtt{cX}$, affects two qubits, and if t is an expression of type qubit $\otimes$ qubit then also $\mathsf{apply}_{\mathtt{cX}}(t)$ is an expression of type qubit$\otimes$qubit. The computation $\mathsf{apply}_{\mathtt{cX}}(a, b)$ is like 'if a is 1 then return $(a, \neg b)$ else return (a, b)', so that the second value returned depends on the first value input. The entanglement occurs because this controlled rotation happens without actually measuring a, and indeed it is reversible. Yet if a is subsequently measured then the controlled rotation appears to have behaved in this way.

The main contribution of this paper is the fact that the relationship between unitary rotations and measurement can be completely described by three simple axioms (Theorem 9), and allocation by two simple axioms (Theorem 11). This simple axiomatization (combined with the unitary groups and commutativity laws) completely characterizes earlier models that are built from operator algebra and functional analysis.

In the remainder of this introduction, we give an informal



Full abstraction for quantum channels

- **Result:** Let $\Lambda, \Gamma : M_n(\mathbb{C}) \to M_k(\mathbb{C})$ be quantum channels. Then $\Lambda = \Gamma$ by Staton's axioms iff $\Lambda = \Gamma$ as morphisms in $L(R(\mathbf{Unitary}))$.
- **In other words:** Every equality of quantum channels can be established using nothing but…
  - Equality of pure maps in $\mathbf{Unitary}$ (or any other ambient base category, such as our free model Π_k — note E4 not needed in this case!).
  - The rig category axioms.
  - The definition of the constructions $L(-)$ and $R(-)$.
- Imminently useful for formalisation of quantum algorithms, and for quantum program semantics.

Want to know more?

PNAS RESEARCH ARTICLE | COMPUTER SCIENCES OPEN ACCESS Check for updates

Free quantum computing

Jacques Carette[a,1], Chris Heunen[b,1,2], Robin Kaarsgaard[c,1], Neil J. Ross[d,1], and Amr Sabry[e,1]

Edited by Peter Shor, Massachusetts Institute of Technology, Cambridge, MA; received May 1, 2025; accepted January 8, 2026

Quantum computing improves substantially on known classical algorithms for various important problems, but the nature of the relationship between quantum and classical computing is not yet fully understood. This relationship can be clarified by free models, that add to classical computing just enough physical principles to represent quantum computing and no more. Here, we develop an axiomatization of quantum computing that replaces the standard continuous postulates with a small number of discrete equations, as well as a free model that replaces the standard linear-algebraic model with a category-theoretical one. The axioms and model are based on reversible classical computing, isolate quantum advantage in the ability to take certain well-behaved square roots, and link to various quantum computing hardware platforms. This approach allows combinatorial optimization, including brute force computer search, to optimize quantum computations. The free model may be interpreted as a programming language for quantum computers, that has the same expressivity and computational universality as the standard model, but additionally allows automated verification and reasoning.