Recursion in circuit description languages

Bert Lindenhovius, Michael Mislove and Vladimir Zamdzhiiev

Department of Computer Science
Tulane University

March 8, 2019
Proto-Quipper-M

- We will consider several variants of a functional programming language called Proto-Quipper-M (renamed to ECLNL in our LICS paper).
  - We wanted to emphasize its dependence on enrichment in the name.
Proto-Quipper-M

- We will consider several variants of a functional programming language called *Proto-Quipper-M* (renamed to ECLNL in our LICS paper).
  - We wanted to emphasize its dependence on enrichment in the name.

- Original language developed by Francisco Rios and Peter Selinger.

- Language is equipped with formal denotational and operational semantics.

- Primary application is in quantum computing, its purpose is to generate quantum circuits, but the language can describe arbitrary string diagrams.

- Original model does not support general recursion.
Overview of this talk

Part 1  Extending the language with general recursion.
  • Soundness result;
  • Adequacy result for the fragment of the language without circuits;

Part 2  Extending the fragment without circuits with recursive types;
  • The resulting language can be regarded as a linear/non-linear extension of FPC;
  • Soundness and adequacy results.

Part 3  The quest for a suitable concrete category for quantum computing.
PQM/ECLNL is used to describe families of morphisms of an arbitrary, but fixed, symmetric monoidal category, which we denote $M$.

Example
If $M = \text{FdCStar}$, the category of finite-dimensional $C^*$-algebras and completely positive maps, then a program in our language is a family of quantum circuits.

Example
$M$ could also be a category of string diagrams which is freely generated.
Circuit Model

Example

Shor’s algorithm for integer factorization may be seen as an infinite family of quantum circuits – each circuit is a procedure for factorizing an $n$-bit integer, for a fixed $n$.

Figure: Quantum Fourier Transform on $n$ qubits (subroutine in Shor’s algorithm).\textsuperscript{1}

\textsuperscript{1}Figure source: https://commons.wikimedia.org/w/index.php?curid=14545612
Syntax of ECLNL calculus

The types of the language:

Types

\[ A, B ::= \alpha | 0 | A + B | I | A \otimes B | A \rightarrow B | !A | \text{Circ}(T, U) \]

Intuitionistic types

\[ P, R ::= 0 | P + R | I | P \otimes R | !A | \text{Circ}(T, U) \]

M-types

\[ T, U ::= \alpha | I | T \otimes U \]

The term language:

Terms

\[ M, N ::= x | I | c | \text{let } x = M \text{ in } N \]

\[ | \square_A M | \text{left}_{A,B} M | \text{right}_{A,B} M | \text{case } M \text{ of } \{\text{left } x \rightarrow N | \text{right } y \rightarrow P\} \]

\[ | * | M; N | \langle M, N \rangle | \text{let } \langle x, y \rangle = M \text{ in } N | \lambda x^A.M | MN \]

\[ | \text{lift } M | \text{force } M | \text{box}_T M | \text{apply}(M, N) | (\bar{I}, C, \bar{V}) \]
Our approach

- Describe an *abstract* categorical model for the same language.
- Describe an abstract categorical model for the language extended with recursion.

**Related work:** Rennela and Staton describe a different circuit description language, called EWire (based on QWire), where they also use enriched category theory.
Linear/Non-Linear models

A Linear/Non-Linear (LNL) model as described by Benton is given by the following data:

- A cartesian closed category $V$.
- A symmetric monoidal closed category $C$.
- A symmetric monoidal adjunction:

$$V \dashv C$$

Remark

An LNL model is a model of Intuitionistic Linear Logic.

Nick Benton. A mixed linear and non-linear logic: Proofs, terms and models. CSL’94
Models of the Enriched Effect Calculus

A model of the Enriched Effect Calculus (EEC) is given by the following data:

- A cartesian closed category $\mathcal{V}$, enriched over itself.

- A $\mathcal{V}$-enriched category $\mathcal{C}$ with powers, copowers, finite products and finite coproducts.

- A $\mathcal{V}$-enriched adjunction:

$$
\begin{array}{ccc}
\mathcal{V} & \xrightarrow{F} & \mathcal{C} \\
\downarrow & & \downarrow \\
\downarrow & & \downarrow \\
\mathcal{C} & \xleftarrow{G} & \mathcal{V}
\end{array}
$$

**Theorem**

*Every LNL model with additives determines an EEC model.*

An abstract model for ECLNL

A model of ECLNL is given by the following data:

1. A cartesian closed category $\mathbf{V}$ together with its self-enrichment $\mathbf{V}$, such that $\mathbf{V}$ has finite $\mathbf{V}$-coproducts.

2. A $\mathbf{V}$-symmetric monoidal closed category $\mathbf{C}$ with underlying category $\mathbf{C}$ such that $\mathbf{C}$ has finite $\mathbf{V}$-coproducts.

3. A $\mathbf{V}$-symmetric monoidal adjunction: $\mathbf{V} \xrightarrow{\perp} \mathbf{C}$, where $(- \odot I)$ denotes the $\mathbf{V}$-copower of the tensor unit in $\mathbf{C}$.

4. A symmetric monoidal category $\mathbf{M}$ and a strong symmetric monoidal functor $E : \mathbf{M} \to \mathbf{C}$.

**Theorem:** Ignoring condition 4, an LNL model canonically induces a model of ECLNL.
Copying and discarding of intuitionistic types

\[ \vdash C \leftarrow \neg \otimes I \]
\[ C(\neg, -) \]

In PQM, any type \( A \) is interpreted as an object \([A]\) in the underlying category \( C \). Terms are interpreted as morphisms in \( C \).

**Theorem**

*For any intuitionistic type \( P \), there exists a canonical isomorphism \( \alpha_P : [P] \to F(P) \).*

Next, define copy and discard morphisms for each intuitionistic type \( P \):

\[ \diamond_P := [P] \xrightarrow{\alpha_P} F(P) \xrightarrow{F1} F1 \xrightarrow{\cong} I \]

\[ \Delta_P := [P] \xrightarrow{\alpha_P} F(P) \xrightarrow{F(id,id)} F((P) \times (P)) \xrightarrow{\cong} F(P) \otimes F(P) \xrightarrow{\alpha_P^{-1} \otimes \alpha_P^{-1}} [P] \otimes [P] \]
Soundness

Theorem (Soundness)

Every abstract model of ECLNL is computationally sound.
Concrete models of ECLNL

The original Proto-Quipper-M model is given by the LNL model: ²

²Thanks to Sam Staton for asking why do we need the Fam construction for this.
Concrete models of ECLNL

The original Proto-Quipper-M model is given by the LNL model: \(^2\) A simpler model for the same language is given by: where in both cases \(\mathcal{C} = [\mathcal{M}^{\text{op}}, \mathsf{Set}]\).

\(^2\)Thanks to Sam Staton for asking why do we need the \textbf{Fam} construction for this.
Concrete models of the base language (contd.)

Fix an arbitrary symmetric monoidal category $\mathcal{M}$. Equipping $\mathcal{M}$ with the free CPO-enrichment yields another concrete (order-enriched) ECLNL model: where $\mathcal{C} = [\mathcal{M}^{\text{op}}, \text{CPO}]$. 
Recursion

Extend the syntax:

\[
\frac{\Phi, x : !A; \emptyset \vdash m : A}{\Phi; \emptyset \vdash \text{rec } x^{!A} m : A} \quad (\text{rec})
\]

Extend the operational semantics:

\[
(C, m[y \leftarrow \text{lift rec } x^{!A} m/x]) \downarrow (C', v) \\
(C, \text{rec } x^{!A} m) \downarrow (C', v)
\]
Abstract model with recursion?

Definition
An endofunctor $T : \mathbf{C} \to \mathbf{C}$ is parametrically algebraically compact, if for every $A \in \text{Ob}(\mathbf{C})$, the endofunctor $A \otimes T(\cdot)$ has an initial algebra and a final coalgebra whose carriers coincide.

Theorem
A categorical model of a linear/non-linear lambda calculus extended with recursion is given by an LNL model:

$$
\begin{array}{c}
\text{V} \\
\downarrow \quad F \\
\text{C}
\end{array}
\quad
\begin{array}{c}
\quad C \\
\downarrow G \\
\text{V}
\end{array}
$$

where $FG$ (or equivalently $GF$) is parametrically algebraically compact $^3$.

$^3$Benton & Wadler. Linear logic, monads and the lambda calculus. LiCS’96.
ECLNL extended with general recursion

Definition
A categorical model of ECLNL extended with general recursion is given by a model of ECLNL, where in addition:

5. The comonad endofunctor:

\[ \mathcal{V} \xrightarrow{\perp} C, \]

is parametrically algebraically compact.
Soundness

Theorem (Soundess)

Every model of ECLNL extended with recursion is computationally sound.
Concrete model of ECLNL extended with recursion

Let $M_*$ be the free $\text{CPO}_\bot !$-enrichment of $M$ and $\overline{M}_* = [M_*^\text{op}, \text{CPO}_\bot !]$ be the associated enriched functor category.

Remark

If $M = 1$, then the above model degenerates to the left vertical adjunction, which is a model of a LNL lambda calculus with general recursion.
Computational adequacy

Theorem
The following LNL model:

\[
\begin{array}{c}
\bot & \cdots & CPO \\
\downarrow & \cdots & \downarrow \\
\bot & \cdots & CPO_\bot!
\end{array}
\]

is computationally adequate at intuitionistic types for the circuit-free fragment of ECLNL.
Recursive types

- The models for Proto-Quipper-M in the previous part seem to support recursive types.
- Main difficulty is on the denotational side.
- How can we copy/discard intuitionistic recursive types?
  - A list of qubits should be linear – cannot copy/discard.
  - A list of natural numbers should be intuitionistic – can implicitly copy/discard.
- For the rest of the talk we focus on the linear/non-linear type structure.
- Recall FPC, a language with coproducts, products, exponentials, and recursive types. How do we design a linear/non-linear FPC?
Adding Recursive Types

Type Variables

Types

Intuitionistic types

Remark

• These types are accompanied by some formation rules, which we omit.
• We use the same type variable for both intuitionistic and general (both linear and intuitionistic) types.
Some useful recursive types

Example
Nat ≡ µX.l + X  (intuitionistic)

Example
List Nat ≡ µX.l + X ⊗ Nat  (intuitionistic)

Example
List Qubit ≡ µX.l + X ⊗ Qubit  (linear)

Example
Stream Qubit ≡ µX.l ⊸ (X ⊗ Qubit)  (linear)

Example
Stream Nat ≡ µX.!(X ⊗ Nat)  (intuitionistic)
We previously extended PQM with a recursion operator:

\[
\Phi, x : !A; \emptyset \vdash m : A \\
\Phi; \emptyset \vdash \text{rec } x^{!A} m : A
\]  

(rec)

Moreover, we extended the operational semantics:

\[
(C, m[\text{lift rec } x^{!A} m/x]) \Downarrow (C', \nu) \\
(C, \text{rec } x^{!A} m) \Downarrow (C', \nu)
\]
Term level recursion

In FPC, a term-level recursion operator may be defined using fold/unfold maps. The same is true for our language.

**Theorem**

The term-level recursion operator for PQM$^4$ is now a derived rule. For a given term $\Phi, z :!A \vdash m : A$, define:

$$\alpha_m^z \equiv \text{lift fold } \lambda x^{!\mu X.(!X \rightarrow A)}.(\lambda z^{!A}.m)(\text{lift (unfold force } x) x)$$

$$\text{rec } z^{!A}.m \equiv (\text{unfold force } \alpha_m^z)\alpha_m^z$$

---

$^4$Bert Lindenhovius, Michael Mislove, Vladimir Zamdzhiev: Enriching a Linear/Non-linear Lambda Calculus: A Programming Language for String Diagrams. LICS 2018
Embedding-projection pairs

**Problem:** How do we interpret recursive types which also contain ! and \(\circ\)?

**Textbook Solution:** CPO-enrichment and embedding-projection pairs.

**Definition**
Given a CPO-enriched category \( \mathcal{C} \), an *embedding-projection* pair is a pair of morphisms \( e : A \to B \) and \( p : B \to A \), such that \( p \circ e = \text{id} \) and \( e \circ p \leq \text{id} \).

**Theorem**
*If \( e \) is an embedding, then it has a unique projection, which we denote \( e^* \).*

**Definition**
The subcategory of \( \mathcal{C} \) with the same objects, but whose morphisms are embeddings is denoted \( \mathcal{C}_e \).
Interpretation of recursive types

In PQM, any type $A$ is interpreted as an object $\llbracket A \rrbracket \in C$. If we equip the language with recursive types, expressions as $\Theta \vdash A$ will be interpreted as functors $\llbracket \Theta \vdash A \rrbracket : C^n \to C$ for some suitable category $C$. Interpreting (closed) recursive types amounts to finding initial (final) (co)algebras of endofunctors induced by these functors.

**Lemma (Adámek)**

Let $C$ be a category with an initial object $\emptyset$ and let $T : C \to C$ be an endofunctor. Assume further that the following $\omega$-diagram

$$
\emptyset \overset{\iota}{\to} T\emptyset \overset{T\iota}{\to} T^2\emptyset \overset{T^2\iota}{\to} \cdots
$$

has a colimit and $T$ preserves it. Then, the induced isomorphism is the initial $T$-algebra.

**Corollary**

In a symmetric monoidal closed category with finite coproducts and $\omega$-colimits, any endofunctor composed from constants, $\otimes$ and $+$ has an initial algebra.
Interpretation of recursive types (contd.)

Theorem (Smyth and Plotkin)

If \( T : \mathbf{C} \to \mathbf{D} \) is a \( \mathbf{CPO} \)-enriched functor and \( \mathbf{C} \) has \( \omega \)-colimits, then \( T \) preserves \( \omega \)-colimits of embeddings. In other words, the restriction \( T_e : \mathbf{C}_e \to \mathbf{D}_e \) is \( \omega \)-continuous.

Theorem

In our categorical model, any \( \mathbf{CPO} \)-endofunctor \( T : \mathbf{C} \to \mathbf{C} \) has an initial \( T \)-algebra, whose inverse is a final \( T \)-coalgebra.

Remark

The above theorem follows directly from results in Fiore’s PhD thesis.
A **CPO**-enriched model

1. A **CPO**-symmetric monoidal closed category $\mathcal{C}$ such that $\mathcal{C}$ has finite **CPO**-coproducts.
2. A **CPO**-symmetric monoidal adjunction:

\[
\begin{array}{ccc}
\mathcal{CPO} & \xrightarrow{\perp} & \mathcal{C}, \\
\downarrow & & \downarrow \\
\mathcal{C}(I, -) & &
\end{array}
\]

3. The category $\mathcal{C}$ is **CPO**$_{\perp I}$-enriched and has $\omega$-colimits.

**Remark**

1. and 3. imply $\mathcal{C}$ has a zero object and we can solve recursive domain equations.
Recursive types for PQM

Using the data from our categorical model:

Using the data from our categorical model:

\[ \text{CPO} \xrightarrow{F} \text{C} \xleftarrow{G} \]

we may solve all required recursive domain equations and interpret all required type expressions \( \Theta \vdash A \) as functors \( \llbracket \Theta \vdash A \rrbracket : \text{C}_e^n \rightarrow \text{C}_e \).

**Remark**

*This follows easily using well-known results from the literature.*

**Problem:** How do we copy/discard the (recursive) intuitionistic types?
Pre-embeddings

Definition
Given two CPO-enriched categories $\mathcal{C}$ and $\mathcal{D}$ and a CPO-functor $T : \mathcal{C} \to \mathcal{D}$, a \textit{pre-embedding} in $\mathcal{C}$ w.r.t $T$ is a morphism $f \in \mathcal{C}$, s.t. $Tf$ is an embedding in $\mathcal{D}$.

Definition
Let $\text{CPO}_{\text{pe}}$ be the subcategory of $\text{CPO}$ with the same objects, but whose morphisms are pre-embeddings w.r.t $F$ in our model.

Example
Every embedding in $\text{CPO}$ is a pre-embedding, but not vice versa. The empty map $\iota : \emptyset \to X$ is a pre-embedding (w.r.t to $F$ in our model), but not an embedding.
Copying and discarding?

Recall that in PQM with basic types, the basis for copying and discarding is given by the canonical iso (for $P$ intuitionistic):

$$\alpha_P : [[P]] \cong F(P)$$

**Problem:** How do we generalise this to work with recursive types, where the interpretation of a type is now a functor?
Interpreting intuitionistic types

**Theorem**

For any intuitionistic type $\Theta \vdash P$ with $n = |\Theta|$, we can find an $\omega$-continuous functor $(\Theta \vdash P) : \text{CPO}^n_{pe} \to \text{CPO}_{pe}$, and moreover a natural isomorphism

$$\alpha_{\Theta \vdash P} : [\Theta \vdash P] \circ F^\times n \Rightarrow F \circ (\Theta \vdash P)$$

diagrammatically (with upper left corner $C^n_{ne}$, and upper right corner $C_e$):
Concrete model

Let $M_*$ be the free $\text{CPO}_{\bot!}$-enrichment of $M$ and $\overline{M}_* = [M_*^{\text{op}}, \text{CPO}_{\bot!}]$ be the associated enriched functor category.

![Diagram]

Remark

If $M = 1$, then the above model degenerates to the left vertical adjunction, which is a model of FPC.
Computational soundness and adequacy

Theorem
The proposed model is computationally sound.

Theorem
The proposed model is computationally adequate at intuitionistic types.
Issues/future work

- Is it possible to work abstractly instead of enriching over cpo’s?
- Dependent types?
- No recursive types yet for the language with circuits;
  - Soundness is likely not a problem, but adequacy is;
  - Issue lies in that the tensor product does not reflect the order;
  - Fundamental issue that our method does not work if $M$ has more than one scalar;
  - Working with the Day tensor does not make life easier.
A category of quantum computing

Consider the framework of a model for PQM with $M$ the category of finite-dimensional algebras:

$$
\begin{aligned}
V & \vdash C \cdot - \otimes I \\
& \downarrow \quad \downarrow \\
V & \quad C(I,-)
\end{aligned}
$$

We aim to find a concrete category $C$ of quantum computing in terms of operators on a Hilbert space for the following reasons:

- Operators on Hilbert spaces are used by physicists to interpret quantum physics;
A category of quantum computing

Consider the framework of a model for PQM with $\mathbf{M}$ the category of finite-dimensional algebras:

\[
\begin{array}{ccc}
\text{V} & \xrightarrow{\mathbf{C}(I, -)} & \text{C} \\
\downarrow & \circ & \downarrow \\
\bot & \circ & \bot \\
\end{array}
\]

We aim to find a concrete category $\mathbf{C}$ of quantum computing in terms of operators on a Hilbert space for the following reasons:

- Operators on Hilbert spaces are used by physicists to interpret quantum physics;
- No longer dealing with the Day tensor, which is complicated;
A category of quantum computing

Consider the framework of a model for PQM with $\mathbf{M}$ the category of finite-dimensional algebras:

$$
\begin{array}{ccc}
\mathcal{V} & \xrightarrow{\perp} & \mathcal{C} \\
\downarrow \quad \downarrow \quad \downarrow \\
\mathcal{C}(I,-) & \quad & \\
\end{array}
$$

We aim to find a concrete category $\mathbf{C}$ of quantum computing in terms of operators on a Hilbert space for the following reasons:

- Operators on Hilbert spaces are used by physicists to interpret quantum physics;
- No longer dealing with the Day tensor, which is complicated;
- A concrete category might allow us to adding dynamic lifting, i.e., including the execution of the quantum circuits;
A category of quantum computing

Consider the framework of a model for PQM with $M$ the category of finite-dimensional algebras:

\[ V \quad \bot \quad C \]

\[ \text{C}(I, -) \]

We aim to find a concrete category $C$ of quantum computing in terms of operators on a Hilbert space for the following reasons:

- Operators on Hilbert spaces are used by physicists to interpret quantum physics;
- No longer dealing with the Day tensor, which is complicated;
- A concrete category might allow us to adding dynamic lifting, i.e., including the execution of the quantum circuits;
- Which might allow us to use a different concept of adequacy, namely a probabilistic version for the language with circuits;
A category of quantum computing

Consider the framework of a model for PQM with $\mathbf{M}$ the category of finite-dimensional algebras:

We aim to find a concrete category $\mathbf{C}$ of quantum computing in terms of operators on a Hilbert space for the following reasons:

- Operators on Hilbert spaces are used by physicists to interpret quantum physics;
- No longer dealing with the Day tensor, which is complicated;
- A concrete category might allow us to adding dynamic lifting, i.e., including the execution of the quantum circuits;
- Which might allow us to use a different concept of adequacy, namely a probabilistic version for the language with circuits;
- Such a concrete category might also support subtyping.
The type of a qubit

- Pure states: $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis for $\mathbb{C}^2$;
The type of a qubit

• Pure states: \( \alpha |0\rangle + \beta |1\rangle \) with \( \alpha, \beta \in \mathbb{C} \) and where \( \{|0\rangle, |1\rangle\} \) is an orthonormal basis for \( \mathbb{C}^2 \);

• (Mixed) states of qubits correspond to density matrices in \( M_2(\mathbb{C}) \);
The type of a qubit

- Pure states: \( \alpha|0\rangle + \beta|1\rangle \) with \( \alpha, \beta \in \mathbb{C} \) and where \( \{|0\rangle, |1\rangle\} \) is an orthonormal basis for \( \mathbb{C}^2 \);
- (Mixed) states of qubits correspond to density matrices in \( \mathbb{M}_2(\mathbb{C}) \);
- Observables correspond to hermitian matrices in \( \mathbb{M}_2(\mathbb{C}) \);
The type of a qubit

- Pure states: $\alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis for $\mathbb{C}^2$;
- (Mixed) states of qubits correspond to density matrices in $M_2(\mathbb{C})$;
- Observables correspond to hermitian matrices in $M_2(\mathbb{C})$;
The type of a qubit

- Pure states: $\alpha |0\rangle + \beta |1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis for $\mathbb{C}^2$;
- (Mixed) states of qubits correspond to density matrices in $\mathbb{M}_2(\mathbb{C})$;
- Observables correspond to hermitian matrices in $\mathbb{M}_2(\mathbb{C})$;

This suggest to assign $\mathbb{M}_2(\mathbb{C})$ to the type of a qubit.
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
- Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
- Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
- Type of conditional branching of systems with types $A$ and $B$: the direct sum $A \oplus B$. 
Finite-dimensional complex algebras

• The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
• Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
• Type of conditional branching of systems with types $A$ and $B$: the direct sum $A \oplus B$.
• Hence it is reasonable to take $\mathcal{M}$ to be the category of finite-dimensional complex algebras.
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(C) \cong M_{2^n}(C)$;
- Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
- Type of conditional branching of systems with types $A$ and $B$: the direct sum $A \oplus B$.
- Hence it is reasonable to take $\mathcal{M}$ to be the category of finite-dimensional complex algebras.
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
- Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
- Type of conditional branching of systems with types $A$ and $B$: the direct sum $A \oplus B$.
- Hence it is reasonable to take $\mathcal{M}$ to be the category of finite-dimensional complex algebras.

$\mathcal{M}$ is monoidal closed. Yet we do not want to take $\mathcal{C} = \mathcal{M}$, because $\mathcal{M}$ has no infinite types such as
- natural number type;
- streams.
Finite-dimensional complex algebras

- The type of a combined system of $n$ qubits is $\bigotimes_{i=1}^{n} M_2(\mathbb{C}) \cong M_{2^n}(\mathbb{C})$;
- Hence a category $\mathcal{M}$ whose objects are the quantum types should contain the complex matrix algebras;
- Type of conditional branching of systems with types $A$ and $B$: the direct sum $A \oplus B$.
- Hence it is reasonable to take $\mathcal{M}$ to be the category of finite-dimensional complex algebras.

$\mathcal{M}$ is monoidal closed. Yet we do not want to take $\mathcal{C} = \mathcal{M}$, because $\mathcal{M}$ has no infinite types such as
  - natural number type;
  - streams.
Moreover, there does not seem to be an adjunction between $\text{Set}$ and $\mathcal{M}$. 
A quantum system can be represented by a Hilbert space $H$, where
- observables are represented by hermitian operators $a : H \to H$;
The Hilbert space framework

A quantum system can be represented by a Hilbert space $H$, where

- observables are represented by hermitian operators $a : H \rightarrow H$;
- states are represented by density operators $d : H \rightarrow H$;
The Hilbert space framework

A quantum system can be represented by a Hilbert space $H$, where

- observables are represented by hermitian operators $a : H \to H$;
- states are represented by density operators $d : H \to H$;
- the expectation value of measuring the observable $a$ when the system is in the state $d$ equals $\text{Tr}(ad)$. 
A quantum system can be represented by a Hilbert space $H$, where

- observables are represented by hermitian operators $a : H \to H$;
- states are represented by density operators $d : H \to H$;
- the expectation value of measuring the observable $a$ when the system is in the state $d$ equals $\text{Tr}(ad)$. 
The Hilbert space framework

A quantum system can be represented by a Hilbert space $H$, where

- observables are represented by hermitian operators $a : H \to H$;
- states are represented by density operators $d : H \to H$;
- the expectation value of measuring the observable $a$ when the system is in the state $d$ equals $\text{Tr}(ad)$.

This suggests that a proper generalization of finite-dimensional algebras are algebras of operators on a Hilbert space.
Given a Hilbert space $H$, the set $B(H)$ of continuous linear operators $H \to H$ is:

- equal to $M_n(\mathbb{C})$ if $H = \mathbb{C}^n$;
Given a Hilbert space $H$, the set $B(H)$ of continuous linear operators $H \to H$ is:

- equal to $M_n(\mathbb{C})$ if $H = \mathbb{C}^n$;
- an algebra over $\mathbb{C}$ where addition and scalar multiplication are defined pointwise, multiplication is defined by composition;
Given a Hilbert space $H$, the set $B(H)$ of continuous linear operators $H \rightarrow H$ is:

- equal to $M_n(\mathbb{C})$ if $H = \mathbb{C}^n$;
- an algebra over $\mathbb{C}$ where addition and scalar multiplication are defined pointwise, multiplication is defined by composition;
- complete in the norm induced by the norm on $H$:
  $$\|a\| = \sup\{\|ah\| : h \in H, \|h\| = 1\}.$$
Given a Hilbert space $H$, the set $B(H)$ of continuous linear operators $H \to H$ is:

- equal to $M_n(\mathbb{C})$ if $H = \mathbb{C}^n$;
- an algebra over $\mathbb{C}$ where addition and scalar multiplication are defined pointwise, multiplication is defined by composition;
- complete in the norm induced by the norm on $H$:
  \[
  \|a\| = \sup \{ \|ah\| : h \in H, \|h\| = 1 \}.
  \]
- equipped with an involution $a \mapsto a^*$ via the inner product on $H$: $\langle k, ak \rangle = \langle a^*k, h \rangle$ for each $h, k \in H$;
Operator algebras

An operator algebra is a subalgebra of $B(H)$, usually closed under the involution and closed with respect to some topology:

- $C^*$-algebras are closed with respect to the topology of uniform convergence;
Operator algebras

An operator algebra is a subalgebra of $B(H)$, usually closed under the involution and closed with respect to some topology:

- $C^*$-algebras are closed with respect to the topology of uniform convergence;
- $W^*$-algebras are closed with respect to the topology of pointwise convergence.
Operator algebras

An *operator algebra* is a subalgebra of $\mathcal{B}(H)$, usually closed under the involution and closed with respect to some topology:

- $C^*$-algebras are closed with respect to the topology of uniform convergence;
- $W^*$-algebras are closed with respect to the topology of pointwise convergence.
Operator algebras

An operator algebra is a subalgebra of $B(H)$, usually closed under the involution and closed with respect to some topology:

- $C^*$-algebras are closed with respect to the topology of uniform convergence;
- $W^*$-algebras are closed with respect to the topology of pointwise convergence.

Example

- Every finite-dimensional operator algebra is isomorphic to some finite direct sums of matrix algebras
Operator algebras

An *operator algebra* is a subalgebra of $B(H)$, usually closed under the involution and closed with respect to some topology:

- $C^*$-algebras are closed with respect to the topology of uniform convergence;
- $W^*$-algebras are closed with respect to the topology of pointwise convergence.

**Example**

- Every finite-dimensional operator algebra is isomorphic to some finite direct sums of matrix algebras
- Every commutative $C^*$-algebra is isomorphic to $C(X)$ for some compact Hausdorff space $X$;
**Operator algebras**

An *operator algebra* is a subalgebra of $B(H)$, usually closed under the involution and closed with respect to some topology:

- C*-algebras are closed with respect to the topology of uniform convergence;
- W*-algebras are closed with respect to the topology of pointwise convergence.

**Example**

- Every finite-dimensional operator algebra is isomorphic to some finite direct sums of matrix algebras
- Every commutative C*-algebra is isomorphic to $C(X)$ for some compact Hausdorff space $X$;
- Every commutative W*-algebra is isomorphic to $L^\infty(X, \mu)$ for some measure space $(X, \mu)$. 
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$. 
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$.
- The tensor product $A \otimes B$ represents the composite system;
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$.
- The tensor product $A \otimes B$ represents the composite system;
- Obtained by taking some completion of the algebraic tensor product of $A$ and $B$;
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$.
- The tensor product $A \otimes B$ represents the composite system;
- Obtained by taking some completion of the algebraic tensor product of $A$ and $B$;
- Several choices of completions possible.
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$.
- The tensor product $A \otimes B$ represents the composite system;
- Obtained by taking some completion of the algebraic tensor product of $A$ and $B$;
- Several choices of completions possible.
- For matrix algebras, there is only one unique choice.
Tensor products and direct sums

Let $A$ and $B$ be two operator algebras representing some quantum systems.

- Conditional branching is represented by the direct sum $A \oplus B$.
- The tensor product $A \otimes B$ represents the composite system;
- Obtained by taking some completion of the algebraic tensor product of $A$ and $B$;
- Several choices of completions possible.
- For matrix algebras, there is only one unique choice.
- The spatial tensor product on $W^*$-algebras, denoted by $\overline{\otimes}$, is regarded as the standard $W^*$-algebra tensor product.
States

Definition

Let $A \subseteq B(H)$ be a unital operator algebra. Then we define an order $\leq$ be the order on $A_{sa} = \{a \in A : a^* = a\}$ by $a \leq b$ if and only if $\langle h, ah \rangle \leq \langle h, bh \rangle$ for each $h \in H$. 

Let $A$ be a unital operator algebra. Then a state is a functional $\omega : A \to \mathbb{C}$ such that

1. $\omega(1_A) = 1$
2. $a \leq b$ implies $\omega(a) \leq \omega(b)$ for each $a, b \in A$.

The states of a C*-algebra $A$ form a convex space; points in its extreme boundary are called pure states. For $A = B(H)$, given a density operator $d$, the map $a \mapsto \text{Tr}(ad)$ defines a state, and the pure states are precisely the functionals $a \mapsto \langle h, ah \rangle$ for unit vectors $h \in H$. 

States

Definition
Let $A \subseteq B(H)$ be a unital operator algebra. Then we define an order $\leq$ be the order on $A_{sa} = \{ a \in A : a^* = a \}$ by $a \leq b$ if and only if $\langle h, ah \rangle \leq \langle h, bh \rangle$ for each $h \in H$.

Definition
Let $A$ be a unital operator algebra. Then a state is a functional $\omega : A \to \mathbb{C}$ such that

1. $\omega(1_A) = 1$;
2. $a \leq b$ implies $\omega(a) \leq \omega(b)$ for each $a, b \in A_{sa}$.
States

Definition
Let $A \subseteq B(H)$ be a unital operator algebra. Then we define an order $\leq$ be the order on $A_{sa} = \{a \in A : a^* = a\}$ by $a \leq b$ if and only if $\langle h, ah \rangle \leq \langle h, bh \rangle$ for each $h \in H$.

Definition
Let $A$ be a unital operator algebra. Then a state is a functional $\omega : A \to \mathbb{C}$ such that

1. $\omega(1_A) = 1$;
2. $a \leq b$ implies $\omega(a) \leq \omega(b)$ for each $a, b \in A_{sa}$.

The states of a $C^*$-algebra $A$ form a convex space; points in its extreme boundary are called pure states. For $A = B(H)$, given a density operator $d$, the map $a \mapsto \text{Tr}(ad)$ defines a state, and the pure states are precisely the functionals $a \mapsto \langle h, ah \rangle$ for unit vectors $h \in H$. 

43 / 55
**-*homomorphisms***

**Definition**

A *-*homomorphism* $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map preserving the multiplication, involution and identity element.
**-homomorphisms**

**Definition**
A ***-homomorphism** $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map preserving the multiplication, involution and identity element.

- ***-homomorphisms** between C*-algebras are automatically continuous.
*-homomorphisms

Definition
A *-homomorphism $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map preserving the multiplication, involution and identity element.

- *-homomorphisms between C*-algebras are automatically continuous.
- *-homomorphisms preserve the order, hence they preserve states, but are not the most general maps preserving states.
*-homomorphisms

Definition
A \( \varphi : A \rightarrow B \) between unital operator algebras \( A \) and \( B \) is a linear map preserving the multiplication, involution and identity element.

- \( \varphi \) is a \( * \)-homomorphism.
- \( * \)-homomorphisms between \( C^* \)-algebras are automatically continuous.
- \( * \)-homomorphisms preserve the order, hence they preserve states, but are not the most general maps preserving states.
- In particular, states themselves are not \( * \)-homomorphisms.
Definition

A positive map $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map that preserves the order.

(Completely) positive maps
(Completely) positive maps

Definition
A positive map $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map that preserves the order.

Problem: positive maps might no longer be positive after the process of adding ancillas.
(Completely) positive maps

**Definition**

A *positive* map $\varphi : A \to B$ between unital operator algebras $A$ and $B$ is a linear map that preserves the order.

**Problem:** positive maps might no longer be positive after the process of adding ancillas.

I.e., $\varphi : A \to B$ positive does not necessarily imply that $M_n(\mathbb{C}) \otimes \varphi : M_n(\mathbb{C}) \otimes A \to M_n(\mathbb{C}) \otimes B$ is positive.
(Completely) positive maps

Definition
A *positive* map $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map that preserves the order.

Problem: positive maps might no longer be positive after the process of adding ancillas. I.e., $\varphi : A \rightarrow B$ positive does not necessarily imply that $M_n(\mathbb{C}) \otimes \varphi : M_n(\mathbb{C}) \otimes A \rightarrow M_n(\mathbb{C}) \otimes B$ is positive.

Definition
A *completely positive* map $\varphi : A \rightarrow B$ between unital operator algebras $A$ and $B$ is a linear map such that $M_n(\mathbb{C}) \otimes \varphi : M_n(\mathbb{C}) \otimes A \rightarrow M_n(\mathbb{C}) \otimes B$ is positive for each $n \in \mathbb{N}$. 
The operator algebraic vs the Hilbert space framework

Theorem (Stinespring)

Let $A$ be a unital $C^*$-algebra. For every completely positive map $\varphi : A \to B(H)$ there is a Hilbert space $K$, a unital $*$-homomorphism $\pi : A \to B(K)$, and a bounded operator $u : H \to K$ such that $\varphi(a) = u^* \pi(a)u$. 
The operator algebraic vs the Hilbert space framework

Theorem (Stinespring)

Let $A$ be a unital $C^*$-algebra. For every completely positive map $\varphi : A \rightarrow B(H)$ there is a Hilbert space $K$, a unital $\ast$-homomorphism $\pi : A \rightarrow B(K)$, and a bounded operator $u : H \rightarrow K$ such that $\varphi(a) = u^* \pi(a) u$.

- The description of (mixed) states is easier in the operator algebraic framework.
The operator algebraic vs the Hilbert space framework

**Theorem (Stinespring)**

Let $A$ be a unital $C^*$-algebra. For every completely positive map $\varphi : A \to B(H)$ there is a Hilbert space $K$, a unital $*$-homomorphism $\pi : A \to B(K)$, and a bounded operator $u : H \to K$ such that $\varphi(a) = u^* \pi(a) u$.

- The description of (mixed) states is easier in the operator algebraic framework.
- Classical systems can also be described by operator algebras, via the functors $C : \text{CptHd}^{\text{op}} \to \text{CStar}$ and $l^{\infty} : \text{Set}^{\text{op}} \to \text{WStar}$, hence the interaction between classical and quantum can be described in one framework;
The operator algebraic vs the Hilbert space framework

Theorem (Stinespring)

Let $A$ be a unital $C^*$-algebra. For every completely positive map $\varphi : A \to B(H)$ there is a Hilbert space $K$, a unital $*$-homomorphism $\pi : A \to B(K)$, and a bounded operator $u : H \to K$ such that $\varphi(a) = u^*\pi(a)u$.

- The description of (mixed) states is easier in the operator algebraic framework.
- Classical systems can also be described by operator algebras, via the functors $C : \text{CptHd}^{\text{op}} \to \text{CStar}$ and $l^\infty : \text{Set}^{\text{op}} \to \text{WStar}$, hence the interaction between classical and quantum can be described in one framework;
- A Hilbert space representation can be regarded as a concrete representation of an operator algebra on a Hilbert space. The possibility to discuss several representations instead of just one gives sometimes a more complete description of the system.
A category of quantum computation

We require the ideal category $\mathbf{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

(1) containing the finite-dimensional algebras $\mathbf{M}$ as a (monoidal) subcategory;
A category of quantum computation

We require the ideal category $\mathbf{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

1. containing the finite-dimensional algebras $\mathbf{M}$ as a (monoidal) subcategory;
2. having coproducts and being monoidal closed;
3. being enriched over a cartesian closed category $\mathbf{V}$ (i.e., $\mathbf{Set}$ or $\mathbf{DCPO}$) such that there is a linear/non-linear adjunction $\mathbf{V} \vdash \mathbf{C}$;
4. to support recursion, we would require $\mathbf{C}$ to be enriched over $\mathbf{DCPO} \bot \mathbf{!}$;
5. have all $\omega$-colimits;
6. the comonad induced by the linear/non-linear adjunction is parametrically algebraically compact.
A category of quantum computation

We require the ideal category $\mathcal{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

(1) containing the finite-dimensional algebras $\mathcal{M}$ as a (monoidal) subcategory;
(2) having coproducts and being monoidal closed;
(3) being enriched over a cartesian closed category $\mathcal{V}$ (i.e., $\text{Set}$ or $\text{DCPO}$) such that there is a linear/non-linear adjunction $\mathcal{V} \dashv \mathcal{C}$,
A category of quantum computation

We require the ideal category \( \mathcal{C} \) to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

1. containing the finite-dimensional algebras \( \mathcal{M} \) as a (monoidal) subcategory;
2. having coproducts and being monoidal closed;
3. being enriched over a cartesian closed category \( \mathcal{V} \) (i.e., \( \text{Set} \) or \( \text{DCPO} \)) such that there is a linear/non-linear adjunction
   \[
   \mathcal{V} \dashv \mathcal{C},
   \]
4. to support recursion, we would require \( \mathcal{C} \) to be enriched over \( \text{DCPO} \);
A category of quantum computation

We require the ideal category $\mathbf{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

1. containing the finite-dimensional algebras $\mathbf{M}$ as a (monoidal) subcategory;
2. having coproducts and being monoidal closed;
3. being enriched over a cartesian closed category $\mathbf{V}$ (i.e., $\text{Set}$ or $\text{DCPO}$) such that there is a linear/non-linear adjunction $\mathbf{V} \dashv \mathbf{C}$;

To support recursion, we would require $\mathbf{C}$ to

4. be enriched over $\text{DCPO}_{\bot!}$;
A category of quantum computation

We require the ideal category $\mathbf{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

1. containing the finite-dimensional algebras $\mathbf{M}$ as a (monoidal) subcategory;
2. having coproducts and being monoidal closed;
3. being enriched over a cartesian closed category $\mathbf{V}$ (i.e., $\mathbf{Set}$ or $\mathbf{DCPO}$) such that there is a linear/non-linear adjunction $\mathbf{V} \downarrow \mathbf{C}$;

To support recursion, we would require $\mathbf{C}$ to

4. be enriched over $\mathbf{DCPO}_\bot$;
5. have all $\omega$-colimits.
A category of quantum computation

We require the ideal category $\mathbf{C}$ to be a category of collections of operators on a Hilbert space satisfying the properties of the linear category in a PQM model:

1. containing the finite-dimensional algebras $\mathbf{M}$ as a (monoidal) subcategory;
2. having coproducts and being monoidal closed;
3. being enriched over a cartesian closed category $\mathbf{V}$ (i.e., $\mathbf{Set}$ or $\mathbf{DCPO}$) such that there is a linear/non-linear adjunction $\mathbf{V} \dashv \mathbf{C}$,

To support recursion, we would require $\mathbf{C}$ to

4. be enriched over $\mathbf{DCPO}_{\perp!}$;
5. have all $\omega$-colimits.
6. the comonad induced by the linear/non-linear adjunction is parametrically algebraically compact.
For categories of operator algebras with *-homomorphisms:

- $\omega$-colimits of finite-dimensional algebras in the category of C*-algebras are the approximately finite-dimensional C*-algebras, or AF-algebras.
ω-colimits

For categories of operator algebras with *-homomorphisms:

- ω-colimits of finite-dimensional algebras in the category of C*-algebras are the \emph{approximately finite-dimensional} C*-algebras, or \emph{AF}-algebras.
- No category containing the AF-algebras will be \textbf{DCPO}_{⊥1}-enriched.
For categories of operator algebras with \(\ast\)-homomorphisms:

- \(\omega\)-colimits of finite-dimensional algebras in the category of C*-algebras are the *approximately finite-dimensional* C*-algebras, or AF-algebras.
- No category containing the AF-algebras will be DCPO\(_{-1}\)-enriched.
- This suggests working with W*-algebras instead of C*-algebras;
W*-algebras with unital normal *-homomorphisms

Let \( \mathcal{WStar} \) be the category of W*-algebras and unital normal *-homomorphism.

**Theorem**

*The (monoidal) functor* \( l^\infty : \mathbf{Set} \to \mathcal{WStar}^{\text{op}} \) *has a right adjoint.*
W*-algebras with unital normal *-homomorphisms

Let $\text{WStar}$ be the category of W*-algebras and unital normal *-homomorphism.

**Theorem**

*The (monoidal) functor* $I^\infty : \text{Set} \rightarrow \text{WStar}^{\text{op}}$ *has a right adjoint.*

**Theorem (Kornell$^5$)**

$\text{WStar}^{\text{op}}$ *is complete and cocomplete, and monoidal closed.*

---

**W*-algebras with unital normal *-homomorphisms**

Let \( \text{WStar} \) be the category of \( W^* \)-algebras and unital normal \( * \)-homomorphism.

**Theorem**

The (monoidal) functor \( l^\infty : \text{Set} \to \text{WStar}^{\text{op}} \) has a right adjoint.

**Theorem (Kornell\(^5\))**

\( \text{WStar}^{\text{op}} \) is complete and cocomplete, and monoidal closed.

- Hence we have a LNL-model;
- Enriched over \( \text{Set} \), but what about \( \text{DCPO} \)?
- Problem: the states are not represented by morphisms in \( \text{WStar}^{\text{op}} \).

---

**W*-algebras with unital normal *-homomorphisms**

Let \( \text{WStar} \) be the category of \( W^* \)-algebras and unital normal \( * \)-homomorphism.

**Theorem**

The (monoidal) functor \( l^\infty : \text{Set} \rightarrow \text{WStar}^{\text{op}} \) has a right adjoint.

**Theorem (Kornell\(^5\))**

\( \text{WStar}^{\text{op}} \) is complete and cocomplete, and monoidal closed.

- Hence we have a LNL-model;
- Enriched over \( \text{Set} \), but what about \( \text{DCPO} \)?
- Problem: the states are not represented by morphisms in \( \text{WStar}^{\text{op}} \).

What about \( W^* \)-algebras with normal completely positive maps?

Order structure on operator algebras

Let \( A \subseteq B(H) \) be a unital operator algebra. Recall the order on \( A_{sa} = \{ a \in A : a^* = a \} \) defined by

\[
a \leq b \iff \langle h, ah \rangle \leq \langle h, bh \rangle \text{ for each } h \in H.
\]

Furthermore, we define \([0, 1]_A = \{ a \in A_{sa} : 0 \leq a \leq 1. \} \).
Order structure on operator algebras

Let $A \subseteq B(H)$ be a unital operator algebra. Recall the order on $A_{sa} = \{ a \in A : a^* = a \}$ defined by

$$a \leq b \iff \langle h, ah \rangle \leq \langle h, bh \rangle \text{ for each } h \in H.$$ 

Furthermore, we define $[0, 1]_A = \{ a \in A_{sa} : 0 \leq a \leq 1. \}.$

**Theorem**

*Let $M$ be a $W^*$-algebra. Then $[0, 1]_M$ is a dcpo.*
Order structure on operator algebras

Let $A \subseteq B(H)$ be a unital operator algebra. Recall the order on $A_{sa} = \{ a \in A : a^* = a \}$ defined by

$$a \leq b \iff \langle h, ah \rangle \leq \langle h, bh \rangle \text{ for each } h \in H.$$ 

Furthermore, we define $[0,1]_A = \{ a \in A_{sa} : 0 \leq a \leq 1 \}$.

**Theorem**

Let $M$ be a $W^*$-algebra. Then $[0,1]_M$ is a dcpo.

Note: $C^*$-algebras $A$ for which $[0,1]_A$ is a dcpo are called monotone complete.
Normal morphisms and DCPO-enrichment

Fact: $W^*$-algebras have an intrinsic topology, called the $\sigma$-weak, ultra-weak or weak* topology. A linear map between $W^*$-algebras that is continuous with respect to this topology is called normal.

---


Normal morphisms and DCPO-enrichment

Fact: $W^*$-algebras have an intrinsic topology, called the $\sigma$-weak, ultra-weak or weak* topology. A linear map between $W^*$-algebras that is continuous with respect to this topology is called normal.

Theorem

Let $\varphi : M \to N$ be a positive map between $W^*$-algebras that is subunital, i.e., $\varphi(1_M) \leq 1_N$. Then $\varphi$ is normal if and only if its restriction to a map $[0,1]^\varphi \to [0,1]^\varphi$ is Scott continuous.

---

Normal morphisms and DCPO-enrichment

Fact: $W^*$-algebras have an intrinsic topology, called the $\sigma$-weak, ultra-weak or weak* topology. A linear map between $W^*$-algebras that is continuous with respect to this topology is called normal.

**Theorem**

Let $\varphi : M \to N$ be a positive map between $W^*$-algebras that is subunital, i.e., $\varphi(1_M) \leq 1_N$. Then $\varphi$ is normal if and only if its restriction to a map $[0,1]^\times M \to [0,1]^\times N$ is Scott continuous.

**Theorem (Cho$^6$, Rennela$^7$)**

The category of $W^*$-algebras and completely positive subunital maps is DCPO$_{\perp!}$-enriched.

---


W*-algebras with completely positive maps

Problem: opposite of the category of W*-algebras with completely positive maps is not monoidal closed with respect to the spatial tensor product (Kornell)
W*-algebras with completely positive maps

Problem: opposite of the category of W*-algebras with completely positive maps is not monoidal closed with respect to the spatial tensor product (Kornell)

Possible solutions:

• Working with monotone complete C*-algebras instead (problem: what is the tensor product?);
W*-algebras with completely positive maps

Problem: opposite of the category of W*-algebras with completely positive maps is not monoidal closed with respect to the spatial tensor product (Kornell)

Possible solutions:

- Working with monotone complete C*-algebras instead (problem: what is the tensor product?);
- Using a different W*-tensor product.
W*-algebras with completely positive maps

Problem: opposite of the category of W*-algebras with completely positive maps is not monoidal closed with respect to the spatial tensor product (Kornell)

Possible solutions:
- Working with monotone complete C*-algebras instead (problem: what is the tensor product?);
- Using a different W*-tensor product.
- Issue: the proof of Kornell makes use of the fact that *-homomorphisms preserve the complete algebraic structure, whereas completely positive maps do not preserve the multiplication.
Operator systems

Definition
Let $H$ be a Hilbert space. Then any \*-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.
Operator systems

Definition
Let $H$ be a Hilbert space. Then any *-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.

Let $\mathcal{C}$ be the opposite of the category of weakly closed operator systems with normal completely positive maps. Then we expect $\mathcal{C}$ to be:

- monoidal closed;
Operator systems

Definition
Let $H$ be a Hilbert space. Then any $*$-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.

Let $\mathbf{C}$ be the opposite of the category of weakly closed operator systems with normal completely positive maps. Then we expect $\mathbf{C}$ to be:

- monoidal closed;
- $\text{DCPO}_\bot$-enriched;
Operator systems

Definition
Let $H$ be a Hilbert space. Then any $*$-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.

Let $\mathbf{C}$ be the opposite of the category of weakly closed operator systems with normal completely positive maps. Then we expect $\mathbf{C}$ to be:

- monoidal closed;
- $\text{DCPO}_\bot$-enriched;
- the monoidal functor $I^\infty : \textbf{Set} \to \mathbf{C}$ has a right adjoint;
Operator systems

Definition
Let $H$ be a Hilbert space. Then any $\ast$-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.

Let $\mathbf{C}$ be the opposite of the category of weakly closed operator systems with normal completely positive maps. Then we expect $\mathbf{C}$ to be:

- monoidal closed;
- $\text{DCPO}_\bot$-enriched;
- the monoidal functor $I^\infty : \text{Set} \to \mathbf{C}$ has a right adjoint;
Operator systems

Definition
Let $H$ be a Hilbert space. Then any $*$-closed subspace $S$ of $B(H)$ containing $1_H$ is called an operator system.

Let $\mathcal{C}$ be the opposite of the category of weakly closed operator systems with normal completely positive maps. Then we expect $\mathcal{C}$ to be:

- monoidal closed;
- $\text{DCPO}_{\bot!}$-enriched;
- the monoidal functor $I^\infty : \text{Set} \to \mathcal{C}$ has a right adjoint;

However, we do not see (yet?) whether there exists a monoidal adjunction between $\text{DCPO}$ and $\mathcal{C}$. 
Summary

• Formulation of abstract models for PQM;
Summary

- Formulation of abstract models for PQM;
- Adding recursion;
Summary

• Formulation of abstract models for PQM;
• Adding recursion;
• Adding recursive types;
Summary

- Formulation of abstract models for PQM;
- Adding recursion;
- Adding recursive types;
- No general adequacy result;
Summary

- Formulation of abstract models for PQM;
- Adding recursion;
- Adding recursive types;
- No general adequacy result;
- In case of quantum circuits, finding a concrete category of quantum computation might help;
Summary

- Formulation of abstract models for PQM;
- Adding recursion;
- Adding recursive types;
- No general adequacy result;
- In case of quantum circuits, finding a concrete category of quantum computation might help;
- Operator systems might form a promising candidate.
Props to the audience

Thank you for your attention.