Compositionality in Categorical Quantum Mechanics

Ross Duncan

University of Strathclyde
Glasgow
Compositionality x 3

• Plain old monoidal category theory:
  —— quantum computing in string diagrams

• Rewriting and substitution:
  —— taking the syntax seriously

• “Quantum theory” as a composite theory
  —— Lack’s composing PROPS

An application: compiling for quantum architecture
1. Quantum theory as string diagrams

How much quantum theory can be expressed as an internal language in some monoidal category?
F.D. Pure state QM

- States: Hilbert spaces
- Compound systems: Tensor product
- Dynamics: Unitary maps
- Non-degenerate measurements: O.N. Bases
F.D. Pure state QM

- States: Hilbert spaces
- Compound systems: Tensor product
- Dynamics: Unitary maps
- Non-degenerate measurements: O.N. Bases

Ambient mathematical framework: $\dagger$-symmetric monoidal categories
F.D. Pure state QM

- States: Hilbert spaces
- Compound systems: Tensor product
- Dynamics: Unitary maps
- Non-degenerate measurements: O.N. Bases

Choose some good generators and relations to capture this stuff

Ambient mathematical framework: $\dagger$-symmetric monoidal categories
Frobenius Algebras

**Theorem:** in $\text{fdHilb}$ orthonormal bases are in bijection with $\dagger$-special commutative Frobenius algebras.

\[
\begin{align*}
\delta &: A \rightarrow A \otimes A \\
\epsilon &: A \rightarrow I \\
\mu &: A \otimes A \rightarrow A \\
\eta &: I \rightarrow A
\end{align*}
\]

Via:

\[
\begin{align*}
\delta &: |a_i\rangle \rightarrow |a_i\rangle \otimes |a_i\rangle \\
\epsilon &: |a_i\rangle \rightarrow 1 \\
\mu &= \delta^\dagger \\
\eta &= \eta^\dagger
\end{align*}
\]

Frobenius Algebras

Represent observables by $\dagger$-special commutative Frobenius algebras:

$$\mu = \begin{array}{c}
\begin{array}{c}
\text{ , } \\
\end{array}
\end{array}, \quad \eta = \begin{array}{c}
\begin{array}{c}
\text{ , }
\end{array}
\end{array}$$

$$\mu^\dagger = \begin{array}{c}
\begin{array}{c}
\text{ , }
\end{array}
\end{array}, \quad \eta^\dagger = \begin{array}{c}
\begin{array}{c}
\text{ . }
\end{array}
\end{array}$$
Frobenius Algebras

Represent observables by $\dagger$-special commutative Frobenius algebras:

$$\mu = \begin{array}{c} \text{tree} \\ \text{box} \end{array}, \quad \eta = \begin{array}{c} \text{disk} \\ \text{line} \end{array}$$
Frobenius Algebras

Represent observables by †-special commutative Frobenius algebras:
Frobenius Algebras

Represent observables by $\dagger$-special commutative Frobenius algebras:

\[
\mu = \begin{array}{c}
\includegraphics[width=1cm]{mu}
\end{array}, \quad \eta = \begin{array}{c}
\includegraphics[width=1cm]{eta}
\end{array}
\]

\[
\mu^\dagger = \begin{array}{c}
\includegraphics[width=1cm]{mu_dagger}
\end{array}, \quad \eta^\dagger = \begin{array}{c}
\includegraphics[width=1cm]{eta_dagger}
\end{array}
\]
Phases

- **Defn**: A phase is a unitary map that commutes with the Frobenius algebra like this:

- **Thm**: The phases form an abelian group
Example: Z-spin

• The following define a Frobenius algebra on the qubit:

\[ \delta : \begin{align*}
|0\rangle &\mapsto |00\rangle \\
|1\rangle &\mapsto |11\rangle
\end{align*} \]

\[ \epsilon : \begin{align*}
|0\rangle &\mapsto 1 \\
|1\rangle &\mapsto 1
\end{align*} \]

• Its group of phases is:

\[ Z_\alpha : \begin{align*}
|0\rangle &\mapsto |0\rangle \\
|1\rangle &\mapsto e^{i\alpha} |1\rangle
\end{align*} \]
Example: Z-spin

\[ \delta : \begin{array}{c|c}
|0\rangle & \mapsto |00\rangle \\
|1\rangle & \mapsto |11\rangle 
\end{array} \quad \epsilon : \begin{array}{c|c}
|0\rangle & \mapsto 1 \\
|1\rangle & \mapsto 1 
\end{array} \]

\[ Z_\alpha = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{pmatrix} \]
Frob. algebras + phases

**Theorem:** Let \( f : n \rightarrow m \) be connected.
Example: X-spin

- The following define a Frobenius algebra on the qubit:
  \[ \delta : \begin{align*}
  |+\rangle &\mapsto |++\rangle \\
  |--\rangle &\mapsto \epsilon : \begin{align*}
  |+\rangle &\mapsto 1 \\
  |--\rangle &\mapsto 1 \\
  \end{align*} \\
  \end{align*} \]

- Its group of phases is:
  \[ X_\beta : \begin{align*}
  |+\rangle &\mapsto |+\rangle \\
  |--\rangle &\mapsto e^{i\beta} |--\rangle \\
  \end{align*} \]
$X$ and $Z$ spins

$\delta : \begin{align*}
|0\rangle & \leftrightarrow |00\rangle \\
|1\rangle & \leftrightarrow |11\rangle
\end{align*}$

$\epsilon : \begin{align*}
|0\rangle & \leftrightarrow 1 \\
|1\rangle & \leftrightarrow 1
\end{align*}$
X and Z spins

δ : $|0\rangle \leftrightarrow |00\rangle$
$|1\rangle \leftrightarrow |11\rangle$

ε : $|0\rangle \leftrightarrow 1$
$|1\rangle \leftrightarrow 1$
X and Z spins

\[ \delta : \begin{array}{l}
|0\rangle \leftrightarrow |00\rangle \\
|1\rangle \leftrightarrow |11\rangle 
\end{array} \quad \epsilon : \begin{array}{l}
|0\rangle \leftrightarrow |1\rangle \\
|1\rangle \leftrightarrow |1\rangle 
\end{array} \]

\[ \begin{array}{ll}
\delta : & |+\rangle \leftrightarrow |++\rangle \\
& |-\rangle \leftrightarrow |--\rangle \\
\epsilon : & |+\rangle \leftrightarrow |1\rangle \\
& |-\rangle \leftrightarrow |1\rangle 
\end{array} \]
Strongly Complementary Observables are Hopf algebras

**Theorem 3**: Two observables are strongly complementary iff they form a Hopf algebra

\[
\begin{align*}
\delta & \circ \epsilon & \mu & \eta \\
\mu & \eta & \delta & \epsilon
\end{align*}
\]

Strongly Complementary Observables are Hopf algebras

**Theorem 3**: Two observables are strongly complementary iff they form a Hopf algebra

Theorem 3: Two observables are strongly complementary iff they form a Hopf algebra.

Strongly Complementary Observables are Hopf algebras

Since we are interested in quantum computing we’ll focus on the X and Z observables.

This is called the **ZX-calculus**.
**ZX-calculus syntax**

\[
\alpha \in [0, 2\pi)
\]

**Defn:** A *diagram* is an undirected open graph generated by the above vertices.
ZX-calculus semantics

\[
\begin{align*}
|0\rangle \otimes n & \rightarrow |0\rangle \otimes m \\
|1\rangle \otimes n & \rightarrow e^{i\alpha} |1\rangle \otimes m \\
|+\rangle \otimes n & \rightarrow |+\rangle \otimes m \\
|\rangle \otimes n & \rightarrow e^{i\alpha} |\rangle \otimes m
\end{align*}
\]
Representing Qubits

\[
\begin{align*}
\left[ \begin{array}{c}
\text{red dot} \\
\text{black dot}
\end{array} \right] &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \\
\left[ \begin{array}{c}
\text{red dot} \\
\text{black dot}
\end{array} \right] \text{ with } \pi &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \\
\left[ \begin{array}{c}
\text{green dot} \\
\text{black dot}
\end{array} \right] &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = |+\rangle \\
\left[ \begin{array}{c}
\text{green dot} \\
\text{black dot}
\end{array} \right] \text{ with } \pi &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = |-\rangle
\end{align*}
\]
Representing Phase shifts

\[
\begin{bmatrix}
\text{Green Node} \\
\alpha
\end{bmatrix} = \begin{pmatrix}
1 & 0 \\
0 & e^{i\alpha}
\end{pmatrix}
\]

\[
\begin{bmatrix}
\text{Red Node} \\
\beta
\end{bmatrix} = \begin{pmatrix}
\cos \frac{\beta}{2} & -i \sin \frac{\beta}{2} \\
-i \sin \frac{\beta}{2} & \cos \frac{\beta}{2}
\end{pmatrix}
\]
Representing Paulis

\[
\begin{bmatrix}
\pi
\end{bmatrix} = \begin{pmatrix}
0 & 1 \\
1 & 0
\end{pmatrix}
\]

\[
\begin{bmatrix}
\pi
\end{bmatrix} = \begin{pmatrix}
1 & 0 \\
0 & -1
\end{pmatrix}
\]
Representing CNot

\[ \land X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]
The ZX-calculus is universal

**Theorem**: Let $U$ be a unitary map on $n$ qubits; then there exists a ZX-calculus term $D$ such that:

$$[D] = U$$
The ZX-calculus is universal

**Theorem**: Let $U$ be a unitary map on $n$ qubits; then there exists a ZX-calculus term $D$ such that:

\[
[D] = U
\]

\[
\begin{align*}
Z_{\beta} & \quad \rightarrow \quad \alpha \\
X_{\alpha} & \quad \rightarrow \quad \beta \\
\cdot \quad \rightarrow \quad \text{Green Red}
\end{align*}
\]
Translating circuits

Steane code encoder:

The $\cdot Z$-gate can be obtained by using a Hadamard gate to transform the target bit of a $\cdot X$ gate.

The encoder

This diagram can be substantially simplified; to do so we rely on some easy lemmas:

Lemma 2.1.
Equations

\[ \alpha + \beta \]  

(spider)

(anti-loop)

(identity)
Equations

Generalised Spider

\[ \alpha + \beta = \alpha \]

(spider)

(anti-loop)

(identity)
Equations
Equations

“Strong Complementarity”

(bialgebra)

(copying)

(hopf)

(π-commute)
\begin{equation}
\alpha + \frac{n \pi}{2} = \alpha
\end{equation}

(colour change)
Equations

A weird one specific to ZX

\( \alpha + \frac{n\pi}{2} \)

\( \alpha \)

(color change)
Example: CNOTS
Example: CNOTS
Graph States

Let $G = (V,E)$ be a simple, undirected graph. Then define:

$$|G\rangle = \bigotimes_{(v,u) \in E} CZ_{vu} \bigotimes_{v \in V} |+\rangle$$

Or in 2D:
STOP!
QUANTO-TIME!
A good reference
A good reference
A good reference
2. Composition in graphical syntax
Composing diagrams

- ZX-calculus terms are arrows in PROP
  — Compose them push-out style

Example 3.12 (The \(\cdot Z\)-gate).

The \(\cdot Z\)-gate can be obtained by using a Hadamard (\(H\)) gate to transform the second qubit of a \(\cdot X\) gate. We obtain a simpler representation using the colour-change rule.

From the presentation of \(\cdot Z\) in the zx-calculus, we can immediately read off that it is symmetric in its inputs. Furthermore, we can prove one of the basic properties of the \(\cdot Z\) gate, namely that it is self-inverse.

Example 3.13 (Bell state).

The following is a zx-calculus term representing a quantum circuit which produces a Bell state, \(|00\rangle + |11\rangle\). We can verify this fact by the equations of the calculus.

The corresponding zx-calculus derivation is a proof of the correctness of this circuit.

The zx-calculus can represent many things which do not correspond to quantum circuits. We now present a criterion to recognise which diagrams do correspond to quantum circuits.
Composing diagrams

- **ZX-calculus terms are arrows in PROP**
  - Tensor them push-out style

Proof. It suffices to show that there are zx-calculus terms for the matrices $Z$, $H$ and $\cdot X$. We have $JHK = H$, $JI - K = Z$ and $JK = \cdot X$ which can be verified by direct calculation. Note that $JK = J\cdot K$, so the presentation of $\cdot X$ is unambiguous.

**Example 3.12** (The $\cdot Z$-gate). The $\cdot Z$-gate can be obtained by using a Hadamard ($H$) gate to transform the second qubit of a $\cdot X$ gate. We obtain a simpler representation using the colour-change rule $H \rightarrow H$. From the presentation of $\cdot Z$ in the zx-calculus, we can immediately read off that it is symmetric in its inputs. Furthermore, we can prove one of the basic properties of the $\cdot Z$ gate, namely that it is self-inverse.

**Example 3.13** (Bell state). The following is a zx-calculus term representing a quantum circuit which produces a Bell state, $|00\rangle + |11\rangle$. We can verify this fact by the equations of the calculus.

The corresponding zx-calculus derivation is a proof of the correctness of this circuit.

The zx-calculus can represent many things which do not correspond to quantum circuits. We now present a criterion to recognise which diagrams do correspond to quantum circuits.
Equational Reasoning
Equational Reasoning
Equational Reasoning

\[
\begin{array}{c}
\ldots \quad \alpha \quad = \quad \alpha \\
\beta \quad (\text{hopf})
\end{array}
\]

\[
\begin{array}{c}
\ldots \quad \beta
\end{array}
\]
Equational Reasoning
Equational Reasoning
Equational Reasoning
Equational Reasoning
Equational Reasoning
Equational Reasoning

Double Pushout Rewriting
Equational Reasoning
Equational Reasoning
Equational Reasoning
3. Composite Theories

I learned all this from Pawel: thanks mate!
**PROPs**

**Defn.** A *PROP* is a strict symmetric monoidal category whose objects are the natural numbers.

**Defn.** A \(\dagger\)-*PROP* is a PROP which has a dagger.

Let \(\mathbb{T}\) be a PROP and let \(\mathcal{C}\) be strict monoidal category.

**Defn:** a \(\mathbb{T}\)-algebra in \(\mathcal{C}\) is a strict monoidal functor from \(\mathbb{T}\) to \(\mathcal{C}\).
PROPs

Syntactic presentation of a PROP:

Generators
symbols with
arity and coarity

Relations
equations between
terms of same type

The coproduct of PROPs is very simple:

$$(\Sigma_1, E_1) + (\Sigma_2, E_2) = (\Sigma_1 + \Sigma_2, E_1 + E_2)$$
Example

The PROP of commutative monoids $\mathcal{M}$

$$\Sigma = \{ \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \}$$

$$E = \{ \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} = \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} , \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} = \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} , \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} = \begin{array}{c} \begin{array}{c} \begin{array}{c} \circ \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \\ \begin{array}{c} \circ \end{array} \end{array} \}$$

The $\mathcal{M}$-algebras in $\mathcal{C}$ are exactly the monoids of $\mathcal{C}$
Example

The PROP of cocommutative comonoids $\mathcal{M}^{\text{op}}$

$\Sigma = \{ \quad \}$

$E = \{ \quad = \quad , \quad = \quad | \quad = \quad , \quad = \} \quad$

The $\mathcal{M}^{\text{op}}$-algebras in $\mathbf{C}$ are the comonoids of $\mathbf{C}$
COMPOSING PROPS

Dedicated to Aurelio Carboni on the occasion of his sixtieth birthday

STEPHEN LACK

Abstract. A PROP is a way of encoding structure borne by an object of a symmetric monoidal category. We describe a notion of distributive law for PROPs, based on Beck’s distributive laws for monads. A distributive law between PROPs allows them to be composed, and an algebra for the composite PROP consists of a single object with an algebra structure for each of the original PROPs, subject to compatibility conditions encoded by the distributive law. An example is the PROP for bialgebras, which is a composite of the PROP for coalgebras and that for algebras.
Composing PROPs

PROPs are monads in a certain (complicated) category. Distributive laws of monads produce composite monads — can do this for PROPs!

\[ \lambda : T; S \Rightarrow S; T \]

This boils down to an equation

for every composable pair.

Composing PROPs

**Proposition:** Given a distributive law
\[ \lambda : \mathbb{1}; S \Rightarrow S; \mathbb{1} \]

Then
\[ f : n \rightarrow m = n \xrightarrow{s} k \xrightarrow{t} m \]

**Proposition:** if
\[ S = (\Sigma_S, E_S) \quad T = (\Sigma_T, E_T) \]

then
\[ S; T = (\Sigma_S + \Sigma_T, E_S + E_T + E_\lambda) \]

Composing PROPs

**Proposition:** Given a distributive law
\[ \lambda : \top; S \Rightarrow S; \top \]

Then
\[ f : n \to m = n \xrightarrow{s} k \xrightarrow{t} m \]

**Proposition:** if
\[ S = (\Sigma_S, E_S) \quad \top = (\Sigma_{\top}, E_{\top}) \]

then
\[ S; \top = (S + \top)/E_\lambda \]

Frobenius Algebras

The PROP \( \mathcal{F} \) of **special commutative Frobenius algebras** arises by a distributive law

\[
\lambda_F : \mathcal{M}^{\text{op}};\mathcal{M} \to \mathcal{M};\mathcal{M}^{\text{op}}
\]

generated by the equations

\begin{align*}
\begin{array}{ccc}
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture}
\end{array}
\end{align*}

The PROP \( \mathcal{B} \) of commutative bialgebras is constructed via a distributive law \( \lambda_B : \mathcal{M}^{\text{op}};\mathcal{C} \to \mathcal{C};\mathcal{M}^{\text{op}} \), generated by the equations

\begin{align*}
\begin{array}{ccc}
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture}
\end{array}
\end{align*}

The PROP \( \mathcal{F} \) of Frobenius algebras is also defined by distributive law, \( \lambda_F : \mathcal{C}^{\text{op}};\mathcal{M} \to \mathcal{M};\mathcal{C}^{\text{op}} \), given by the equations

\begin{align*}
\begin{array}{ccc}
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} \\
\begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
  \draw (0,1) -- (1,0) -- (2,1);
\end{tikzpicture} & = & \begin{tikzpicture}
  \draw (0,0) -- (1,1) -- (2,0);
\end{tikzpicture}
\end{array}
\end{align*}
Phases

Let $G$ be an abelian group; define the PROP $G^\times$ by

$$\Sigma = \{ g : 1 \to 1 \mid g \in G \} \quad E = \{ g \circ h = gh \}$$

Quotient $\mathbb{F} + G^\times$ by the equations

\begin{align*}
g & \Rightarrow g \\
(\text{P})
\end{align*}
Frob. algebras with phases

Recall $\mathbb{F}$ is itself a composite $\mathbb{M};\mathbb{M}^{\text{op}}$ so we can view $\mathbb{F}G$ as an *iterated* distributive law for $\mathbb{M};G^{\times};\mathbb{M}^{\text{op}}$.

This yields a factorisation:

$$f = n \xrightarrow{\nabla} m \xrightarrow{g} m \xrightarrow{\Delta} n'$$

So $\mathbb{F}G$ is the PROP of Frob.algs. with *phases*. 
Bialgebras

The PROP \( \mathbb{B} \) of bialgebras arises by a distributive law
\[
\lambda_B : \mathbb{M} ; \mathbb{M}^{\text{op}} \rightarrow \mathbb{M}^{\text{op}} ; \mathbb{M}
\]
generated by the equations

\[
\begin{align*}
\text{red} &= \text{green} \\
\text{green} &= \text{red} \\
\text{green} &= \text{green} \\
\text{green} &= \Box
\end{align*}
\]

Can do the same for Hopf algebras.
Two Frobenius Algebras?

We can form the coproduct i.e. non-interacting Frobenius algebras with phases.

Factorisation:

\[ f = n \xrightarrow{g_1} d_1 \xrightarrow{h_1} d_2 \xrightarrow{g_2} d_3 \xrightarrow{h_2} \cdots \xrightarrow{g_k} m \]
Sad Face :(  

**Theorem:** \( \mathcal{F} \) does not arise as a distributive law \( \lambda : \mathcal{F}G; \mathcal{F}H \Rightarrow \mathcal{F}H; \mathcal{F}G \)

**Proof:** Recall we need:

\[
\begin{array}{c}
n \xrightarrow{t} k \xrightarrow{s} m \\
\downarrow \\
n \xrightarrow{s'} k' \xrightarrow{t'} m
\end{array}
\]

for every composable pair — including the phase groups

RD + Kevin Dunne, “Interacting Frobenius Algebras are Hopf”, LiCS 2016.
But the news is still pretty good

- No distributive law for ZX-calculus
  — no nice normal forms for the full language
  — this would have been very surprising!

- **But** nice normal forms for every subtheory.
  — the monochrome theory = spiders
  — the phase-free theory = $\mathbb{Z}_2$-matrices
  — the Clifford fragment = ???

- This will be enough for some interesting applications!
4. Compiling

Oh look, category theory can do something useful!
Circuit Perspective
Circuit Perspective

Inputs

Outputs
Circuit Perspective

Inputs

Outputs

\[ \delta \rightarrow \alpha \rightarrow \gamma \rightarrow \beta \]
Circuit Perspective
Circuit Perspective

\[ \alpha \quad \beta \quad \gamma \]

\[ \delta \]
??? Perspective

\[ \delta \quad \gamma \quad \alpha \quad \beta \]
???: Perspective

\[
\delta \quad \gamma
\]

\[
\alpha \quad \beta
\]
Hopf algebra expression
???, Perspective

Hopf algebra normal form
MBQC Perspective
MBQC Perspective

Physical qubits
MBQC Perspective

Prepared qubits
MBQC Perspective

Prepared qubits

Measured qubits
Prepared qubits

\[ \delta \]

\[ \gamma \]

\[ \alpha \]

\[ \beta \]

Any ZX-calculus term can be interpreted as an MBQC in this way.
NQIT Perspective
NQIT Perspective
NQIT Perspective

Few qubit ion traps
Few qubit ion traps

Optical interconnect
NQIT Perspective

Optical interconnect

Few qubit ion traps
NQIT perspective(?)

• What about determinism?
  — unknown in general
  — use standard techniques for specific examples

• What are the trade-offs?
  — non-Clifford gates vs physical qubits
  — circuit depth vs complexity of entanglement