Monoidal Theories and Graph Rewriting in Quantum Computing

Aleks Kissinger Lucas Dixon

October 17, 2009

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces
- FdHilb is a "nice" (dagger, symmetric, compact closed...) monoidal category

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces
- FdHilb is a "nice" (dagger, symmetric, compact closed...) monoidal category
- As such, we have lots of coherence results (Mac Lane, Kelly and Laplaza, Joyal and Street)

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces
- FdHilb is a "nice" (dagger, symmetric, compact closed...) monoidal category
- As such, we have lots of coherence results (Mac Lane, Kelly and Laplaza, Joyal and Street)
- Most importantly, we have a graphical notation

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces
- FdHilb is a "nice" (dagger, symmetric, compact closed...) monoidal category
- As such, we have lots of coherence results (Mac Lane, Kelly and Laplaza, Joyal and Street)
- Most importantly, we have a graphical notation
- Applications to quantum states and processes motivates the study of graphical, monoidal theories in categories like FdHilb

Overview

- The majority of quantum computing is expressed in (finite-dimensional) Hilbert spaces
- FdHilb is a "nice" (dagger, symmetric, compact closed...) monoidal category
- As such, we have lots of coherence results (Mac Lane, Kelly and Laplaza, Joyal and Street)
- Most importantly, we have a graphical notation
- Applications to quantum states and processes motivates the study of graphical, monoidal theories in categories like FdHilb
- To do graphical proofs (esp. automatic ones), we need the correct notion of graph rewriting, as well as sound, efficient algorithms

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.
  - **State evolutions**: Unitary maps ($U^\dagger U = 1$ and $UU^\dagger = 1$)

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.
  - **State evolutions**: Unitary maps ($U^\dagger U = 1$ and $UU^\dagger = 1$)
  - **Observables**: Self-adjoint ($O = O^\dagger$) linear maps

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.
  - **State evolutions**: Unitary maps ($U^\dagger U = 1$ and $UU^\dagger = 1$)
  - **Observables**: Self-adjoint ($O = O^\dagger$) linear maps
  - **Measurement**: Sets of projections, summing to the identity

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.
  - **State evolutions**: Unitary maps ($U^\dagger U = 1$ and $UU^\dagger = 1$)
  - **Observables**: Self-adjoint ($O = O^\dagger$) linear maps
  - **Measurement**: Sets of projections, summing to the identity
  - **Composite states**: tensor product $v_1 \otimes v_2$

Hilbert Space Quantum Mechanics

- Pure state quantum mechanics has:
  - **States**: Elements of a Hilbert space, $v \in \mathcal{H}$
    - In finite dimensions, just think of these as plain old vector spaces with a dot product.
  - **State evolutions**: Unitary maps ($U^\dagger U = 1$ and $UU^\dagger = 1$)
  - **Observables**: Self-adjoint ($O = O^\dagger$) linear maps
  - **Measurement**: Sets of projections, summing to the identity
  - **Composite states**: tensor product $v_1 \otimes v_2$
- Mixed state quantum mechanics has generalisations of the above. We won't talk about that.

Entanglement and the Tensor

- For our purposes, take $\otimes$ to be the Kronecker product:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Entanglement and the Tensor

- For our purposes, take $\otimes$ to be the Kronecker product:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

- For Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, we can construct $\mathcal{H}_1 \otimes \mathcal{H}_2 = \operatorname{span}\{v \otimes u : v \in \mathcal{H}_1, u \in \mathcal{H}_2\}$.
  - $\dim(\mathcal{H}_1 \otimes \mathcal{H}_2) = \dim \mathcal{H}_1 \cdot \dim \mathcal{H}_2$

Entanglement and the Tensor

- Some states $w \in \mathcal{H}_1 \otimes \mathcal{H}_2$ can be written as $v \otimes u$ for $v \in \mathcal{H}_1, u \in \mathcal{H}_2$. These states are called *separable*.

Entanglement and the Tensor

- Some states $w \in \mathcal{H}_1 \otimes \mathcal{H}_2$ can be written as $v \otimes u$ for $v \in \mathcal{H}_1, u \in \mathcal{H}_2$. These states are called *separable*.
- ...but most can't. These are called *entangled*. They can be expressed as some sum $\sum v_i \otimes u_i$ and are very important for doing lots of "quantum-like" stuff like teleportation.

Entanglement and the Tensor

- Some states $w \in \mathcal{H}_1 \otimes \mathcal{H}_2$ can be written as $v \otimes u$ for $v \in \mathcal{H}_1, u \in \mathcal{H}_2$. These states are called *separable*.
- ...but most can't. These are called *entangled*. They can be expressed as some sum $\sum v_i \otimes u_i$ and are very important for doing lots of "quantum-like" stuff like teleportation.
- The Hilbert space $\mathcal{Q} := \mathbb{C}^2$ is called the space of *qubits*.

Entanglement and the Tensor

- Some states $w \in \mathcal{H}_1 \otimes \mathcal{H}_2$ can be written as $v \otimes u$ for $v \in \mathcal{H}_1, u \in \mathcal{H}_2$. These states are called *separable*.
- ...but most can't. These are called *entangled*. They can be expressed as some sum $\sum v_i \otimes u_i$ and are very important for doing lots of "quantum-like" stuff like teleportation.
- The Hilbert space $\mathcal{Q} := \mathbb{C}^2$ is called the space of *qubits*.
- We write the standard basis of $\mathcal{Q}$ in "ket" notation, as $|0\rangle, |1\rangle$. Also, $|ij\rangle$ is shorthand for $|i\rangle \otimes |j\rangle$.

Monoidal Categories

- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism

Monoidal Categories

- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is

Monoidal Categories

- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is
  - *Dagger* if it has an involutive monoidal functor $(-)^\dagger$

Monoidal Categories

- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is
  - *Dagger* if it has an involutive monoidal functor $(-)^\dagger$
  - *Braided* if there is a monoidal natural isomorphism $\sigma_{A,B} : A \otimes B \cong B \otimes A$

Monoidal Categories

- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is
  - *Dagger* if it has an involutive monoidal functor $(-)^\dagger$
  - *Braided* if there is a monoidal natural isomorphism $\sigma_{A,B} : A \otimes B \cong B \otimes A$
  - *Symmetric* if it is braided and $\sigma\sigma = 1$

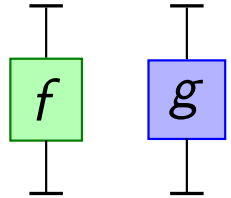
Monoidal Categories

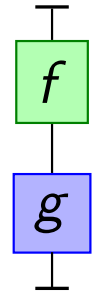
- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is
  - *Dagger* if it has an involutive monoidal functor $(-)^\dagger$
  - *Braided* if there is a monoidal natural isomorphism $\sigma_{A,B} : A \otimes B \cong B \otimes A$
  - *Symmetric* if it is braided and $\sigma\sigma = 1$
- $(\mathrm{FdHilb}, \otimes, \mathbb{C}, (-)^\dagger)$ is a dagger symmetric monoidal category

Monoidal Categories

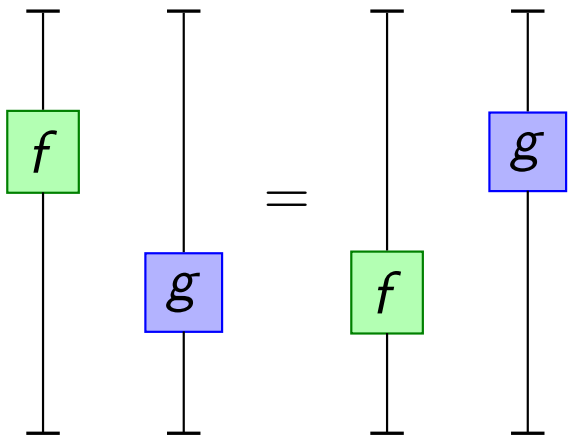
- A *monoidal category* is a category $\mathcal{V}$ equipped with a bifunctor $(- \otimes -)$ that is associative and unital, up to isomorphism
- A monoidal category is
  - *Dagger* if it has an involutive monoidal functor $(-)^\dagger$
  - *Braided* if there is a monoidal natural isomorphism $\sigma_{A,B} : A \otimes B \cong B \otimes A$
  - *Symmetric* if it is braided and $\sigma\sigma = 1$
- $(\mathrm{FdHilb}, \otimes, \mathbb{C}, (-)^\dagger)$ is a dagger symmetric monoidal category
- Other examples include Rel, nCob, *Span*($\mathcal{C}$), *Csp*($\mathcal{C}$), group representation categories, projective spaces, ...

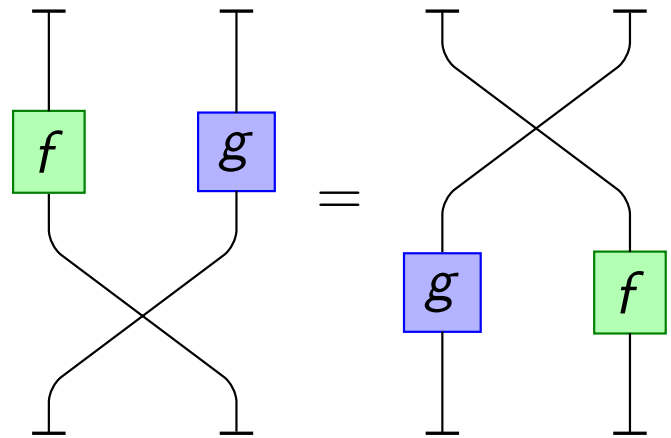
Graphical Representation

$f \otimes g :=$ 

$g \circ f :=$ 

- We can express the bifunctoriality of $\otimes$ and the naturality of σ as follows:



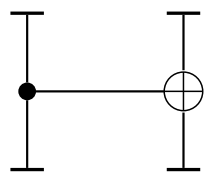


Quantum Circuits

- Any †-SMC provides a notion of *unitarity* ($U^\dagger U = 1$ and $UU^\dagger = 1$)
- We can think of unitary maps as the quantum analogy to reversible logic gates
- Lets look at one motivating example:

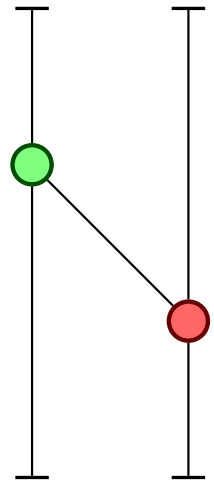
$$\mathrm{CNOT} :: |00\rangle \to |00\rangle, |01\rangle \to |01\rangle, |10\rangle \to |11\rangle, |11\rangle \to |10\rangle$$

- Usually drawn like this:



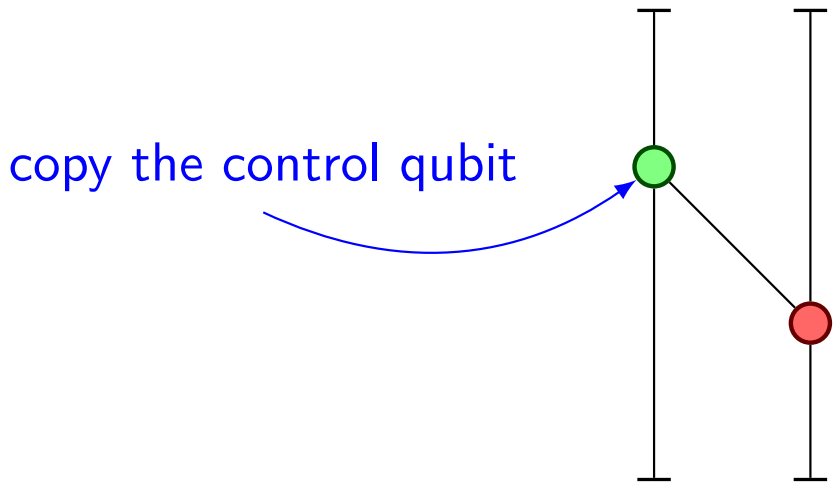
More Primitive

- Lets break CNOT into its primitive, (co)algebraic components



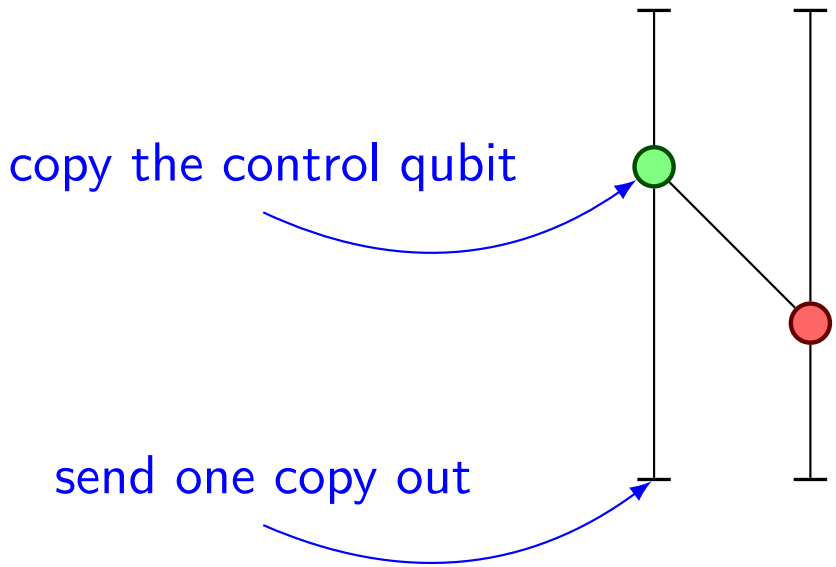
More Primitive

- Lets break CNOT into its primitive, (co)algebraic components



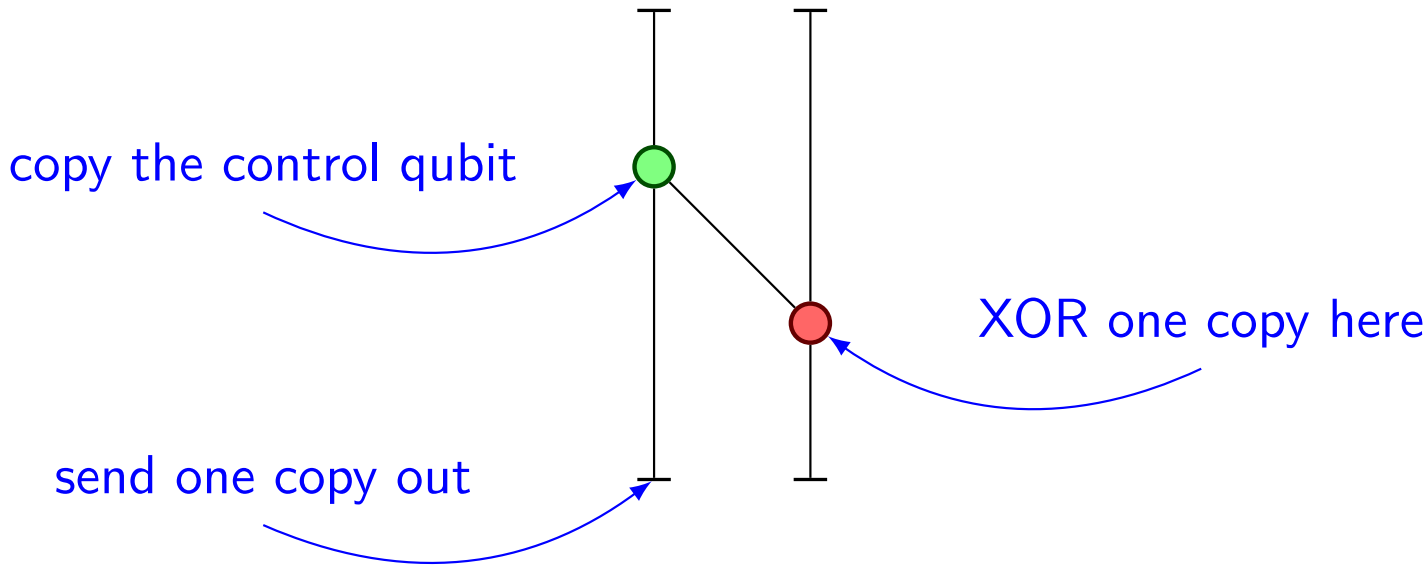
More Primitive

- Lets break CNOT into its primitive, (co)algebraic components



More Primitive

- Lets break CNOT into its primitive, (co)algebraic components

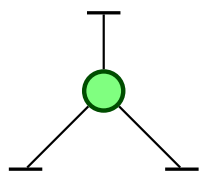


Classical Structures

- A chosen basis is like some classical data embedded in the system.
- What can we do with classical data?
  - Copy and delete!

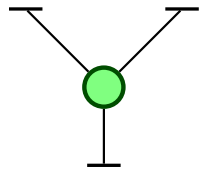
$$\delta_Z : \mathcal{Q} \to \mathcal{Q} \otimes \mathcal{Q} :: |i\rangle \mapsto |ii\rangle \qquad \epsilon_Z : \mathcal{Q} \to \mathbb{C} :: |i\rangle \mapsto 1$$

- Graphically:

$\delta_Z :=$  $\qquad \epsilon_Z :=$ 

Classical Structures

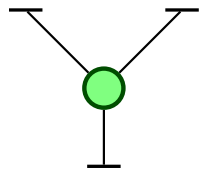
- $(-)^\dagger$ flips everything upside-down:

$\delta_Z^\dagger :=$ 

$\epsilon_Z^\dagger :=$ 

Classical Structures

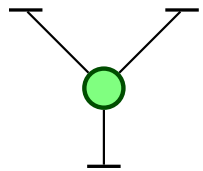
- $(-)^\dagger$ flips everything upside-down:

$\delta_Z^\dagger :=$  $\epsilon_Z^\dagger :=$ 

- $\delta_Z^\dagger$ is a multiplication w.r.t. $\otimes$. Furthermore, it is associative, commutative, and has unit $\epsilon_Z^\dagger$.

Classical Structures

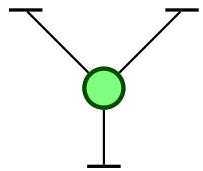
- $(-)^\dagger$ flips everything upside-down:

$\delta_Z^\dagger :=$  $\epsilon_Z^\dagger :=$ 

- $\delta_Z^\dagger$ is a multiplication w.r.t. $\otimes$. Furthermore, it is associative, commutative, and has unit $\epsilon_Z^\dagger$.
- Therefore, we say $(\delta_Z^\dagger, \epsilon_Z^\dagger)$ is an *internal commutative monoid in FdHilb*

Classical Structures

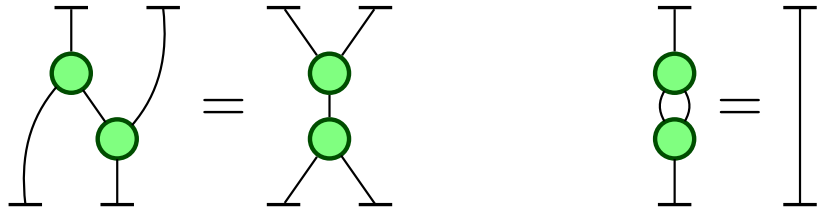
- $(-)^\dagger$ flips everything upside-down:

$\delta_Z^\dagger :=$  $\epsilon_Z^\dagger :=$ 

- $\delta_Z^\dagger$ is a multiplication w.r.t. $\otimes$. Furthermore, it is associative, commutative, and has unit $\epsilon_Z^\dagger$.
- Therefore, we say $(\delta_Z^\dagger, \epsilon_Z^\dagger)$ is an *internal commutative monoid in FdHilb*
- Likewise, (δ_Z, ϵ_Z) is an *internal cocommutative comonoid*

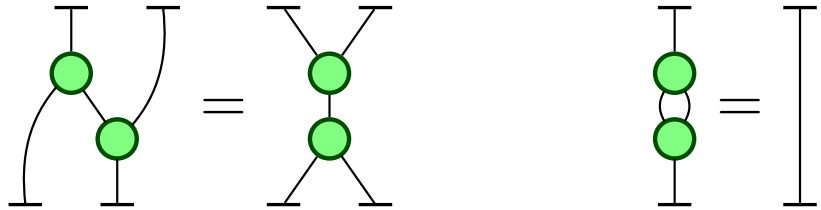
Spiders

- (δ_Z, ϵ_Z) interacts with $(\delta_Z^\dagger, \epsilon_Z^\dagger)$ in the following way:

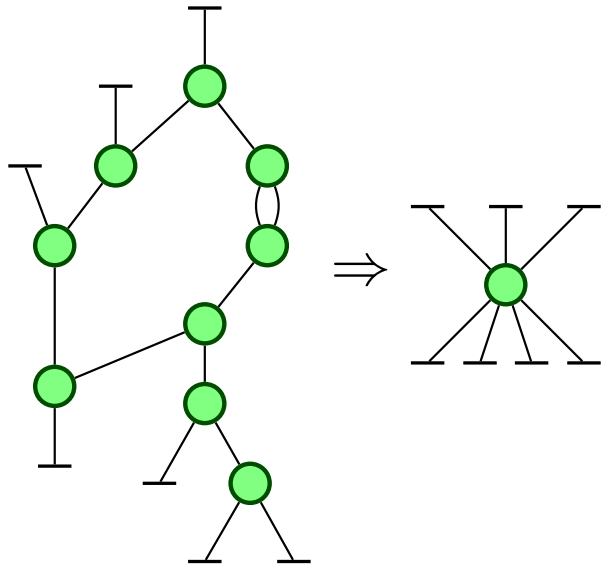


Spiders

- (δ_Z, ϵ_Z) interacts with $(\delta_Z^\dagger, \epsilon_Z^\dagger)$ in the following way:



- Terms of these things are uniquely determined by number of inputs and outputs. As a result, we write connected graphs:



Rewriting Spiders

- A convenient way to express the spider theorem is with *pattern graph rewrites*.

Rewriting Spiders

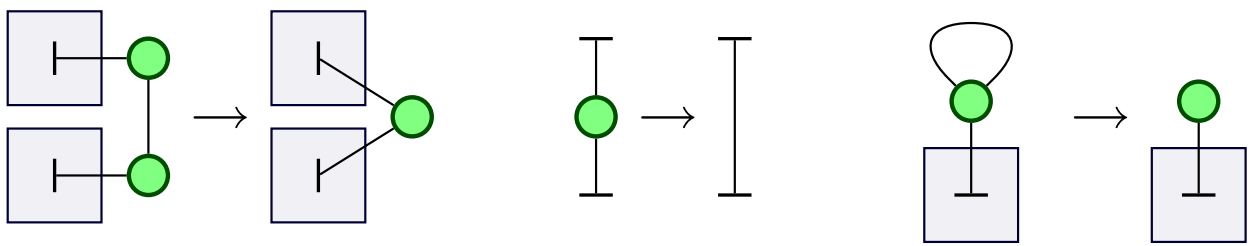
- A convenient way to express the spider theorem is with *pattern graph rewrites*.
- These rules contain "!-boxes", which designate vertices that may be copied an arbitrary number of times

Rewriting Spiders

- A convenient way to express the spider theorem is with *pattern graph rewrites*.
- These rules contain "!-boxes", which designate vertices that may be copied an arbitrary number of times
- A copy on the left is reflected by a copy on the right

Rewriting Spiders

- A convenient way to express the spider theorem is with *pattern graph rewrites*.
- These rules contain "!-boxes", which designate vertices that may be copied an arbitrary number of times
- A copy on the left is reflected by a copy on the right
- The spider theorem:

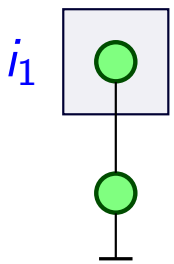


More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:

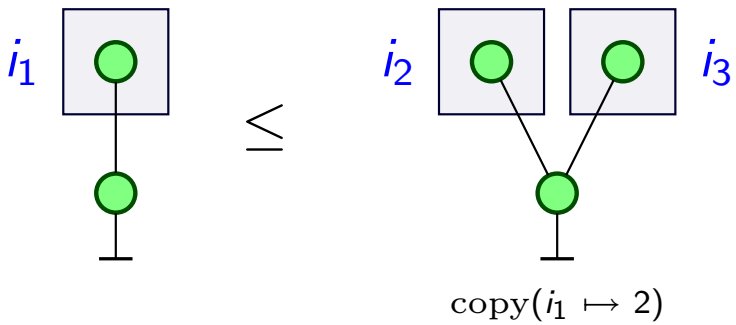
More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



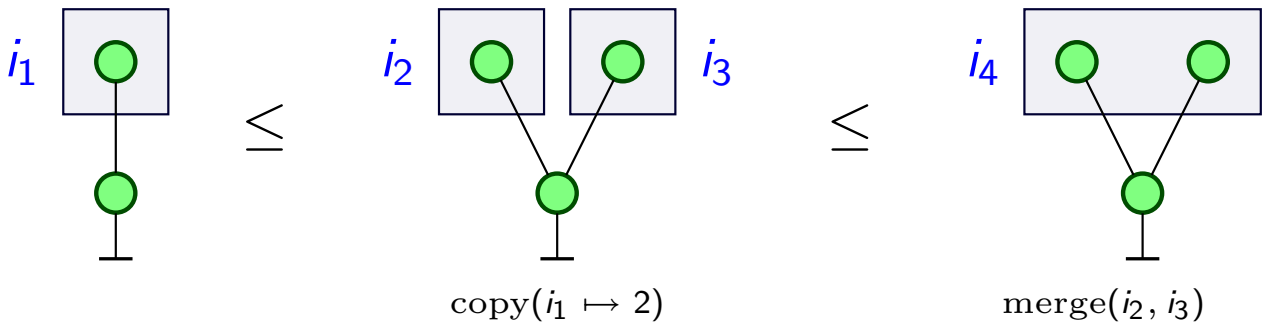
More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



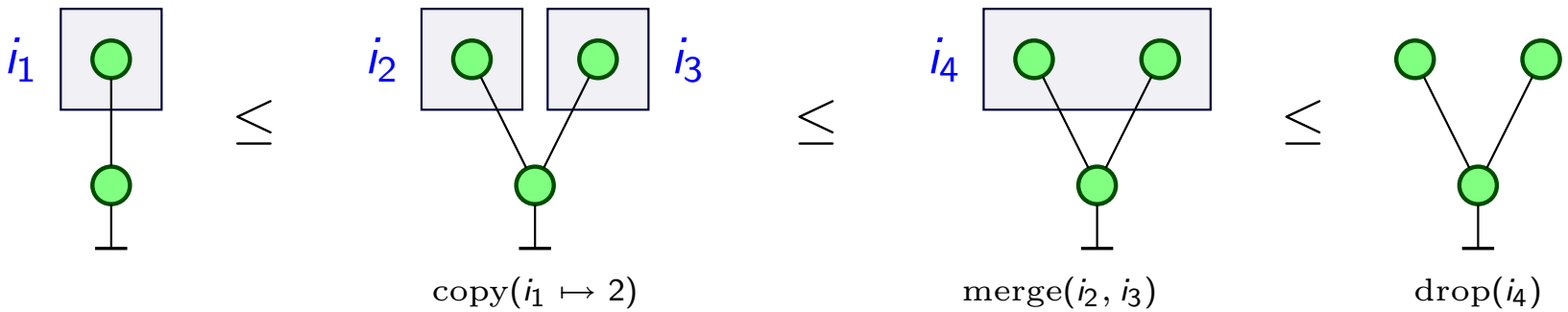
More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



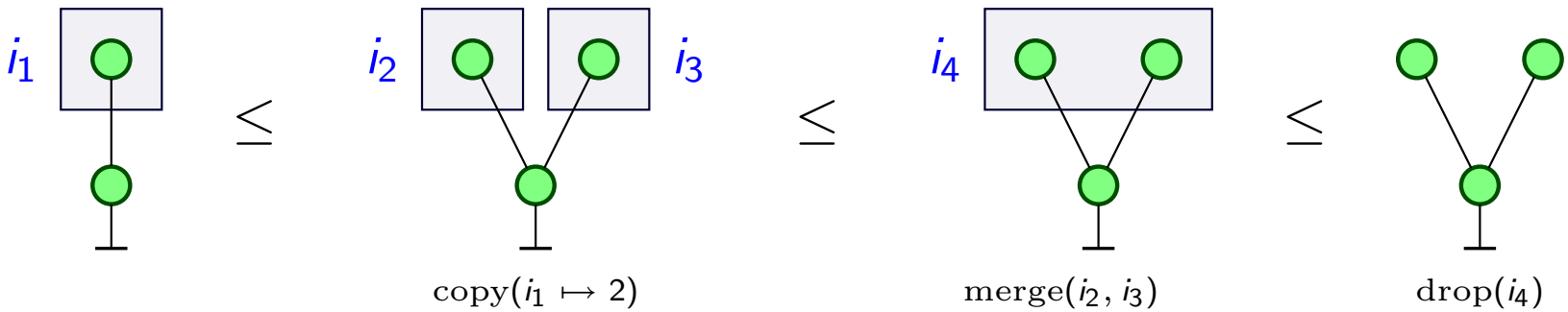
More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



More on !-boxes

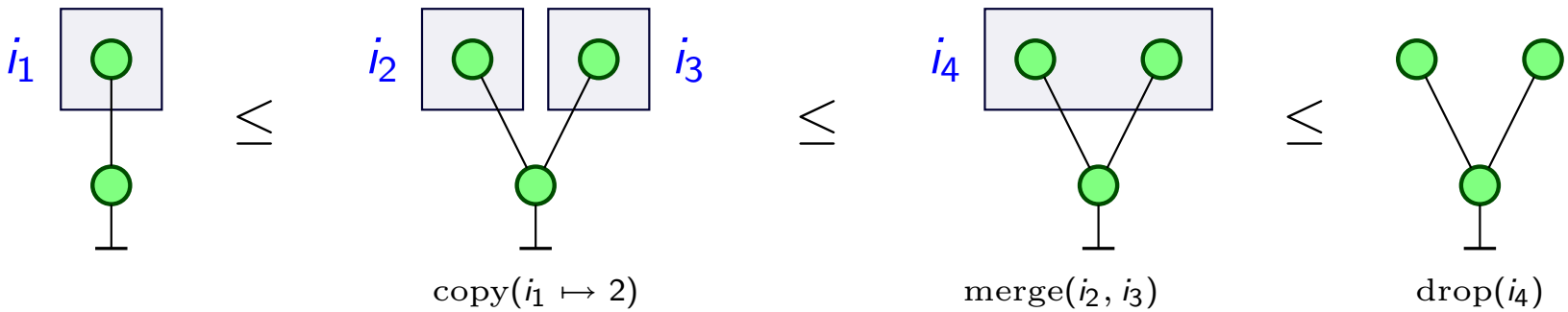
- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



- Naive rewriting algorithm is to expand all combinations of !-boxes, bounded by the size of the target

More on !-boxes

- $G_1 \leq G_2$ iff G_2 can be obtained from G_1 by applying one of three operations:



- Naive rewriting algorithm is to expand all combinations of !-boxes, bounded by the size of the target
- Better algorithm expands lazily during matching

Another colour

- Let (δ_X, ϵ_X) copy and delete this basis:

$$|+\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right) \qquad |-\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)$$

Another colour

- Let (δ_X, ϵ_X) copy and delete this basis:

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

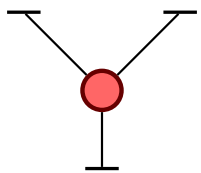
- These two bases are called *mutually unbiased*. They are eigenvectors of *maximally non-commuting observables*. Think position and momentum

Another colour

- Let (δ_X, ϵ_X) copy and delete this basis:

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \qquad |-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

- These two bases are called *mutually unbiased*. They are eigenvectors of *maximally non-commuting observables*. Think position and momentum
- We can depict (δ_X, ϵ_X) with red dots, and note $\delta_X^\dagger$ is the XOR



$:: |00\rangle \mapsto |0\rangle,\ |01\rangle \mapsto |1\rangle,\ |10\rangle \mapsto |1\rangle,\ |11\rangle \mapsto |0\rangle$

Interactions, and a Rewrite Theory

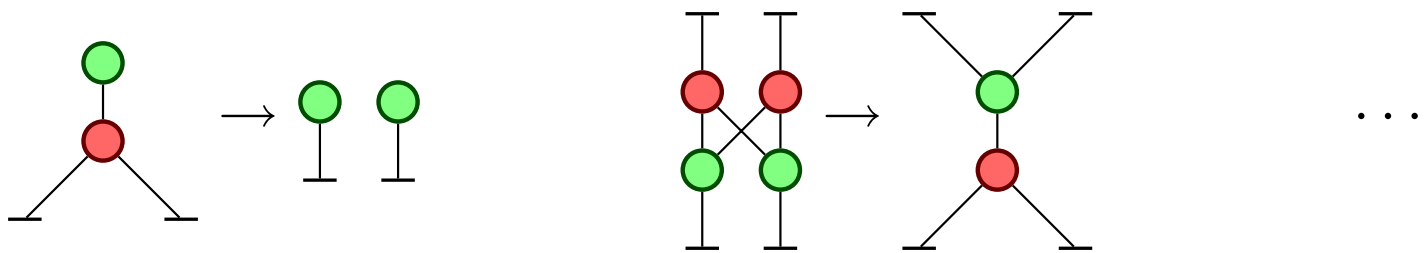
- Red and green are both special frobenius algebras, but they also interact

Interactions, and a Rewrite Theory

- Red and green are both special frobenius algebras, but they also interact
- For instance, the quadruple $(\delta_Z, \epsilon_Z, \delta_X^\dagger, \epsilon_X^\dagger)$ forms another kind of monoidal gadget, called a *Hopf algebra*

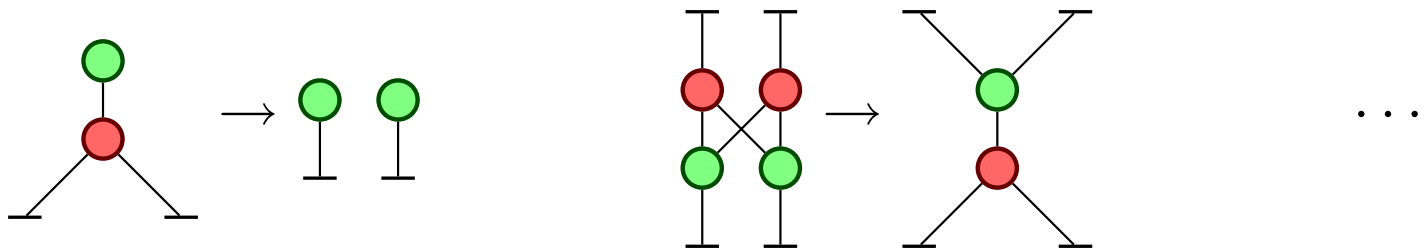
Interactions, and a Rewrite Theory

- Red and green are both special frobenius algebras, but they also interact
- For instance, the quadruple $(\delta_Z, \epsilon_Z, \delta_X^\dagger, \epsilon_X^\dagger)$ forms another kind of monoidal gadget, called a *Hopf algebra*
- This yields some more basic rewrites.



Interactions, and a Rewrite Theory

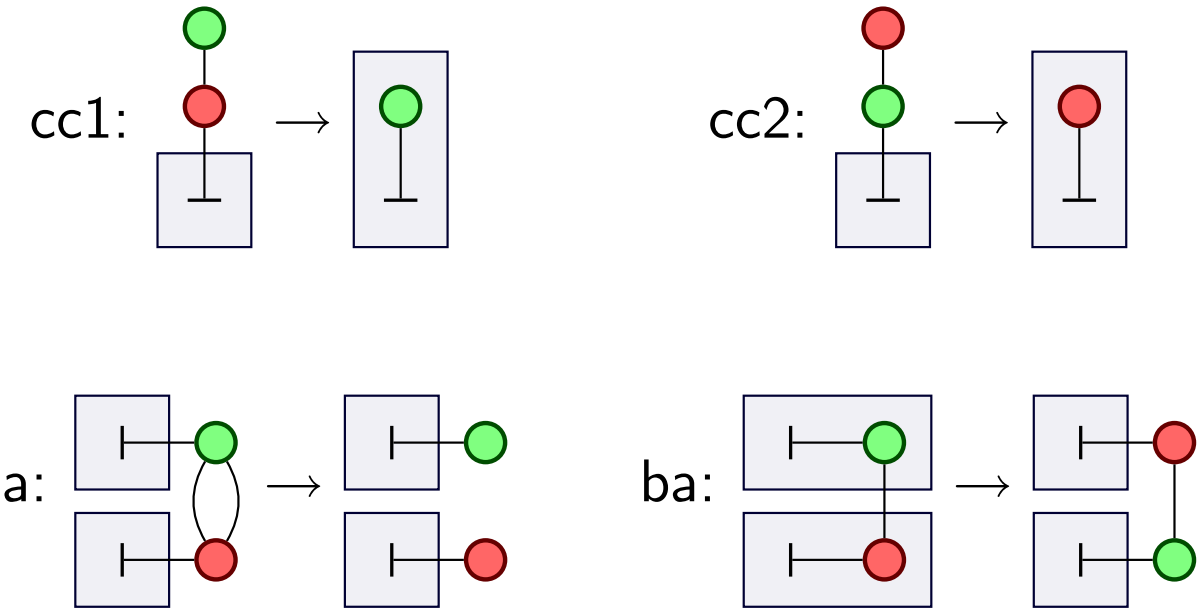
- Red and green are both special frobenius algebras, but they also interact
- For instance, the quadruple $(\delta_Z, \epsilon_Z, \delta_X^\dagger, \epsilon_X^\dagger)$ forms another kind of monoidal gadget, called a *Hopf algebra*
- This yields some more basic rewrites.



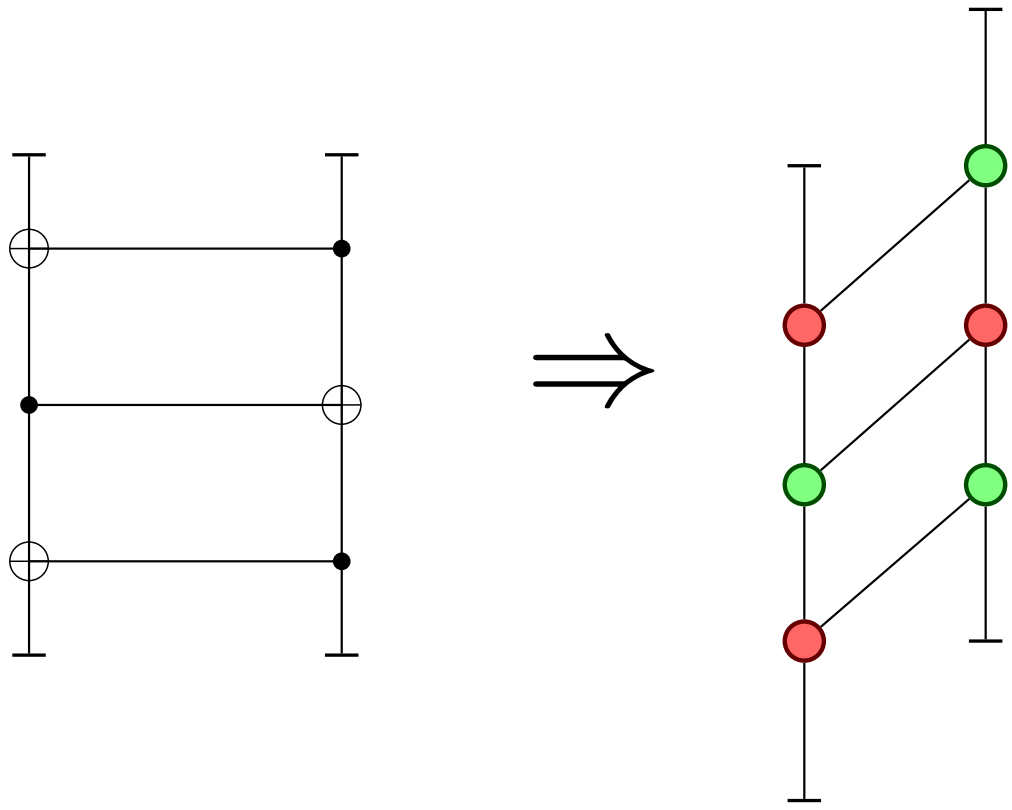
- We can apply pattern rewrites to pattern graphs, so we get...

Interactions, and a Rewrite Theory

- **Derived** pattern rewrites:



Doin' the Swap



Conclusions

- Quantum information has an abstract graphical language, phrased in your favourite flavour of monoidal category

Conclusions

- Quantum information has an abstract graphical language, phrased in your favourite flavour of monoidal category
- Frobenius algebras pop up everywhere (classical data, entangled states, TQFT's)

Conclusions

- Quantum information has an abstract graphical language, phrased in your favourite flavour of monoidal category
- Frobenius algebras pop up everywhere (classical data, entangled states, TQFT's)
- These yield spider theorems, necessitate pattern rewriting

Conclusions

- Quantum information has an abstract graphical language, phrased in your favourite flavour of monoidal category
- Frobenius algebras pop up everywhere (classical data, entangled states, TQFT's)
- These yield spider theorems, necessitate pattern rewriting
- We have a tool for doing automatic rewriting, already being used in study of entangled states (numerically and with rewrites)

Conclusions

- Quantum information has an abstract graphical language, phrased in your favourite flavour of monoidal category
- Frobenius algebras pop up everywhere (classical data, entangled states, TQFT's)
- These yield spider theorems, necessitate pattern rewriting
- We have a tool for doing automatic rewriting, already being used in study of entangled states (numerically and with rewrites)
- Future work: polish the algorithm, more theories, automatic critical pair analysis

Thanks!

- This is joint work with
  - Bob Coecke
    http://www.comlab.ox.ac.uk/people/bob.coecke/
  - Ross Duncan
    http://www.comlab.ox.ac.uk/people/ross.duncan/
  - Lucas Dixon
    http://homepages.inf.ed.ac.uk/ldixon/
- Check it out at
  - http://dream.inf.ed.ac.uk/projects/quantomatic/