Diagrammatic Methods for the Specification and Verification of Quantum Algorithms

William Zeng

Quantum Group
Department of Computer Science
University of Oxford

Quantum Programming and Circuits Workshop
IQC, University of Waterloo
June, 2015

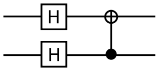
http://willzeng.com/shared/qcircuitworkshop.pdf

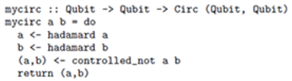
Introduction

- Problem: What are appropriate abstractions for describing quantum algorithms?

Low Level
High Level

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle \qquad \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$





```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

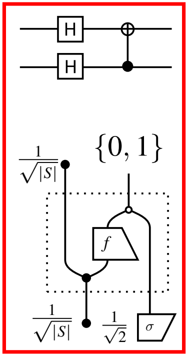
Green et al. arXiv 1304.3390
Wecker & Svore arXiv:1402.4467

Introduction

- Problem: What are appropriate abstractions for describing quantum algorithms?

Low Level
High Level

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle \qquad \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix}$$



```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

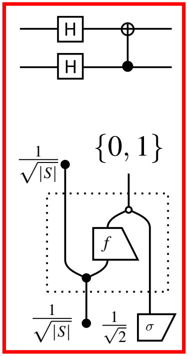
- Quantum Circuits 2.0

Green et al. arXiv 1304.3390
Wecker & Svore arXiv:1402.4467

Introduction

- Problem: What are appropriate abstractions for describing quantum algorithms?

$$i\hbar\frac{\partial}{\partial t}|\psi\rangle = \hat{H}|\psi\rangle \qquad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$



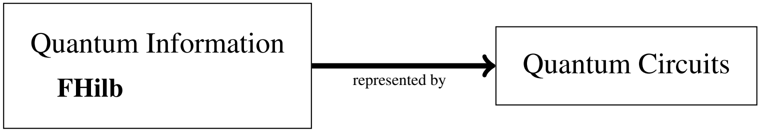
```
mycirc :: Qubit -> Qubit -> Circ (Qubit, Qubit)
mycirc a b = do
  a <- hadamard a
  b <- hadamard b
  (a,b) <- controlled_not a b
  return (a,b)
```

- Quantum Circuits 2.0

$\Rightarrow$

Green et al. arXiv 1304.3390
Wecker & Svore arXiv:1402.4467

Introduction



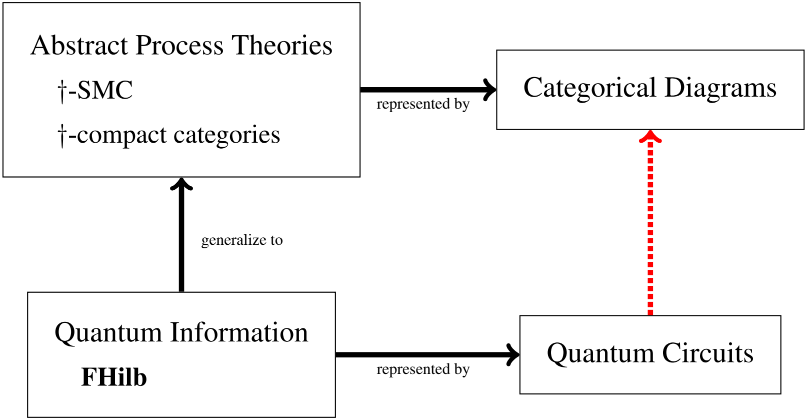
Selinger arXiv 0908.3347

Introduction

$$\begin{tikzcd}
\boxed{\begin{array}{l}\text{Abstract Process Theories}\\ \quad \dagger\text{-SMC}\\ \quad \dagger\text{-compact categories}\end{array}} \arrow[r, "\text{represented by}"'] & \boxed{\text{Categorical Diagrams}} \\
\boxed{\begin{array}{l}\text{Quantum Information}\\ \quad \mathbf{FHilb}\end{array}} \arrow[u, "\text{generalize to}"'] \arrow[r, "\text{represented by}"'] & \boxed{\text{Quantum Circuits}}
\end{tikzcd}$$

Selinger arXiv 0908.3347

Introduction



Selinger arXiv 0908.3347

Introduction

$$\begin{tikzcd}
\boxed{\begin{array}{l}\text{Abstract Process Theories} \\ \quad \dagger\text{-SMC} \\ \quad \dagger\text{-compact categories}\end{array}} \arrow[r, "\text{represented by}"'] & \boxed{\text{Categorical Diagrams}} \\
\boxed{\begin{array}{l}\text{Quantum Information} \\ \quad \mathbf{FHilb}\end{array}} \arrow[u, "\text{generalize to}"'] \arrow[r, "\text{represented by}"'] & \boxed{\text{Quantum Circuits}} \arrow[u, dashed]
\end{tikzcd}$$

Selinger arXiv 0908.3347

Overview

- The Framework: Circuit Diagrams 2.0
  - bases · copying/deleting · groups/representations · complementarity · oracles

Overview

- The Framework: Circuit Diagrams 2.0
  - bases · copying/deleting · groups/representations · complementarity · oracles
- Example 1. Generalized Deutsch-Jozsa algorithm
- Example 2. The quantum GROUPHOMID algorithm

Overview

- The Framework: Circuit Diagrams 2.0
  - bases · copying/deleting · groups/representations · complementarity · oracles
- Example 1. Generalized Deutsch-Jozsa algorithm
- Example 2. The quantum GROUPHOMID algorithm
- Overview of other results.
  - algorithms · locality · foundations
- Outlook.

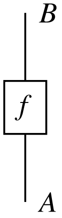
Quantum circuits 1.0

A *category* $\mathbf{C}$ is $\begin{cases} \text{a set of systems } A, B \in \mathrm{Ob}(\mathbf{C}) \\ \text{a set of processes } f : A \to B \in \mathrm{Arr}(\mathbf{C}) \end{cases}$

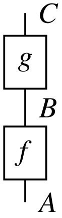
Quantum circuits 1.0

$$\text{A } \textit{category } \mathbf{C} \text{ is } \begin{cases} \text{a set of systems } A, B \in \mathrm{Ob}(\mathbf{C}) \\ \text{a set of processes } f : A \to B \in \mathrm{Arr}(\mathbf{C}) \end{cases}$$

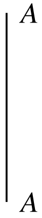
$f : A \to B :=$



$g \circ f :=$

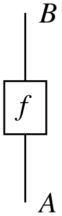


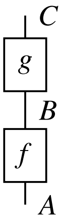
$\mathrm{id}_A :=$

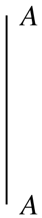


Quantum circuits 1.0

A *category* $\mathbf{C}$ is $\begin{cases} \text{a set of systems } A, B \in \mathrm{Ob}(\mathbf{C}) \\ \text{a set of processes } f : A \to B \in \mathrm{Arr}(\mathbf{C}) \end{cases}$

$f : A \to B :=$


$g \circ f :=$


$\mathrm{id}_A :=$


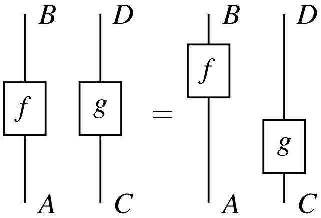
These are sequential processes.

The framework

A *monoidal category* $\mathbf{C}$ has $\begin{cases} \textit{cat. tensor}\ (-\otimes-) : \mathbf{C}\times\mathbf{C}\to\mathbf{C} \\ \text{a } \textit{unit object}\ I \in \mathrm{Ob}(\mathbf{C}) \end{cases}$

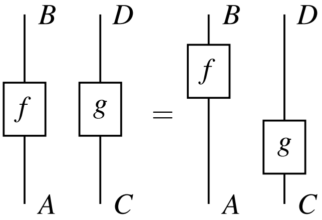
The framework

A *monoidal category* $\mathbf{C}$ has $\begin{cases} \textit{cat. tensor } (-\otimes -) : \mathbf{C}\times\mathbf{C}\to\mathbf{C} \\ \text{a } \textit{unit object } I \in \mathrm{Ob}(\mathbf{C}) \end{cases}$

$f \otimes g :=$  $\qquad \mathrm{id}_I :=$

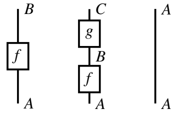
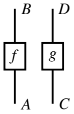
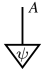
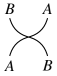
The framework

A *monoidal category* $\mathbf{C}$ has $\begin{cases} \textit{cat. tensor}\ (-\otimes -) : \mathbf{C}\times\mathbf{C}\to\mathbf{C} \\ \text{a } \textit{unit object}\ I \in \mathrm{Ob}(\mathbf{C}) \end{cases}$

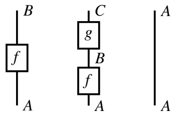
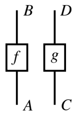
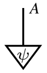
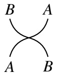
$f \otimes g :=$  $\mathrm{id}_I :=$

These are parallel processes.

Sym. Mon. Cats. & quantum circuits

category	
monoidal category	$f \otimes g :=$  $\mathrm{id}_I :=$
states	$\lvert\psi\rangle :=$ 
symmetric monoidal categories	

Sym. Mon. Cats. & quantum circuits

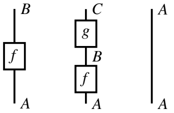
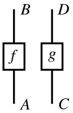
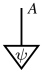
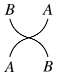
category	
monoidal category	$f \otimes g :=$  $\mathrm{id}_I :=$
states	$\|\psi\rangle :=$ 
symmetric monoidal categories	

Quantum Computation

- **FHilb**: Sym. Mon. Cat.
- Ob(**FHilb**) = f.d. Hilbert Spaces
- Arr(**FHilb**) = linear maps
- $\otimes$ is the tensor product
- $I = \mathbb{C}$
- States are $|\psi\rangle : \mathbb{C} \to \mathcal{H}$

Abramsky & Coecke arXiv 0808.1023

Sym. Mon. Cats. & quantum circuits

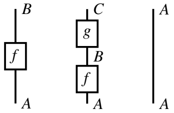
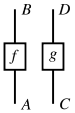
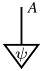
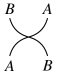
category	
monoidal category	$f \otimes g :=$  $\mathrm{id}_I :=$
states	$\lvert\psi\rangle :=$ 
symmetric monoidal categories	

FHilb : Sym. Mon. Cat.

Ob(**FHilb**) = f.d. Hilbert Spaces

Arr(**FHilb**) = linear maps

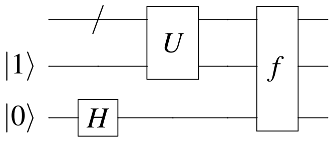
Sym. Mon. Cats. & quantum circuits

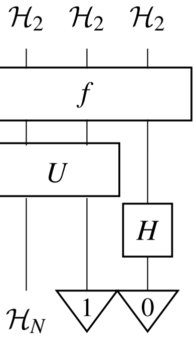
category	
monoidal category	$f \otimes g :=$  $\mathrm{id}_I :=$
states	$\|\psi\rangle :=$ 
symmetric monoidal categories	

FHilb : Sym. Mon. Cat.

Ob(**FHilb**) = f.d. Hilbert Spaces

Arr(**FHilb**) = linear maps





The dagger

A *dagger functor* $\dagger : \mathbf{C} \to \mathbf{C}$ s.t.

$$\left(f^\dagger\right)^\dagger = f \qquad (1)$$

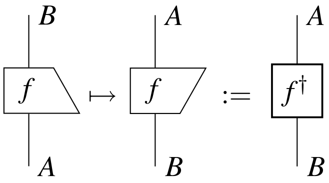
$$(g \circ f)^\dagger = f^\dagger \circ g^\dagger \qquad (2)$$

$$\mathrm{id}_A^\dagger = \mathrm{id}_H \qquad (3)$$

FHilb is a dagger category with the usual adjoint.

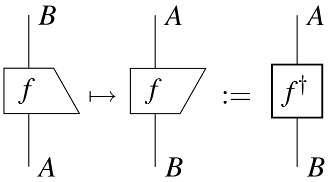
The dagger

A *dagger functor* $\dagger : \mathbf{C} \to \mathbf{C}$

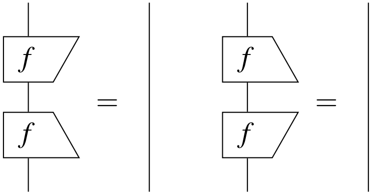


The dagger

A *dagger functor* $\dagger : \mathbf{C} \to \mathbf{C}$

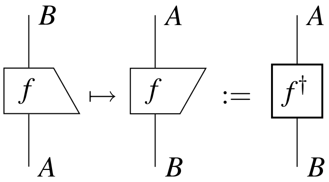


Unitarity:

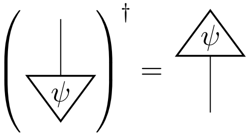


The dagger

A *dagger functor* $\dagger : \mathbf{C} \to \mathbf{C}$

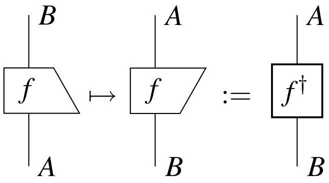


On states:

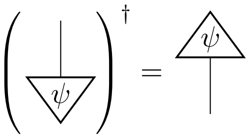


The dagger

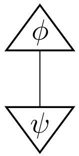
A *dagger functor* $\dagger : \mathbf{C} \to \mathbf{C}$



On states:



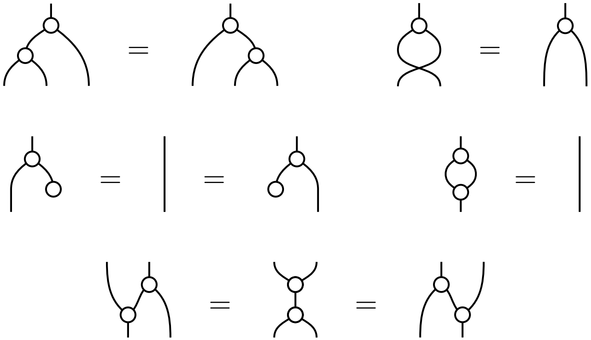
$|\phi\rangle \circ \langle\psi| = \langle\phi|\psi\rangle =$



This is a *scalar* $\langle\phi|\psi\rangle : \mathbb{C} \to \mathbb{C}$ or $I \to I$ in general and admits a generalized Born rule.

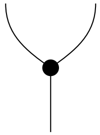
Bases

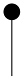
A †-special Frobenius algebra (A, ,) obeys:



Bases

Given a finite set S, we use the following diagrams to represent the ‘copying’ and ‘deleting’ functions:

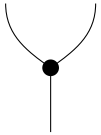


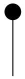


$$S \xrightarrow{\Delta} S \times S \qquad\qquad S \xrightarrow{\epsilon} 1$$

Bases

Given a finite set S, we use the following diagrams to represent the ‘copying’ and ‘deleting’ functions:





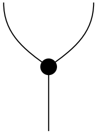
$$S \xrightarrow{\Delta} S \otimes S \qquad\qquad S \xrightarrow{\epsilon} \mathbb{C}$$

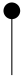
$$|s\rangle \mapsto |s\rangle \otimes |s\rangle \qquad\qquad |s\rangle \mapsto 1$$

We treat these as linear maps acting on a free vector space, whose basis is S.

Bases

Given a finite set S, we use the following diagrams to represent the 'copying' and 'deleting' functions:

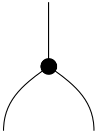


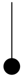


$$S \xrightarrow{\Delta} S \otimes S \qquad S \xrightarrow{\epsilon} \mathbb{C}$$

$$|s\rangle \mapsto |s\rangle \otimes |s\rangle \qquad |s\rangle \mapsto 1$$

We treat these as linear maps acting on a free vector space, whose basis is S.

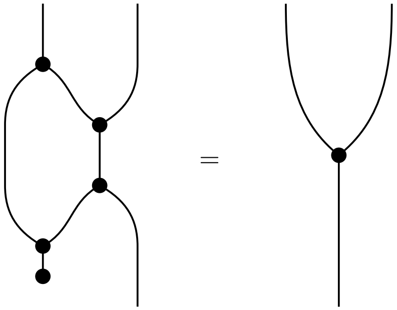




$$|s\rangle \otimes |t\rangle \mapsto \delta_{s,t}|s\rangle \qquad 1 \mapsto \sum_s |s\rangle$$

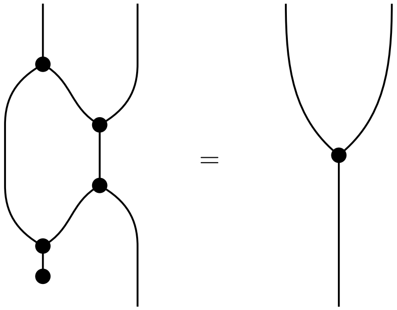
Bases and Topology

These linear maps form a †-special commutative Frobenius algebra. Their composites are determined entirely by their connectivity, e.g.:



Bases and Topology

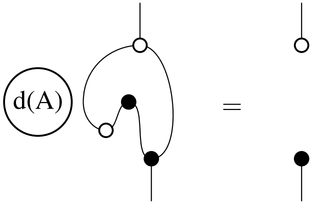
These linear maps form a †-special commutative Frobenius algebra. Their composites are determined entirely by their connectivity, e.g.:



- **[Coecke et al. 0810.0812]** †-(special) commutative Frobenius algebras on objects in **FHilb** are eqv. to orthogonal (orthonormal) bases.
- **[Evans et al. 0909.4453]** †-(special) commutative Frobenius algebras on objects in **Rel** are eqv. to groupoids.

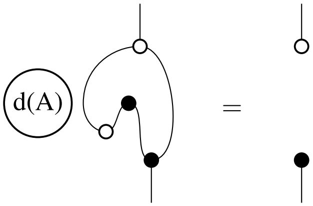
Complementarity

- **[Coecke & Duncan 0906.4725]:** Two †-SCFA's on the same object are complementary when:



Complementarity

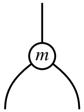
- **[Coecke & Duncan 0906.4725]:** Two †-SCFA's on the same object are complementary when:



- This is the Hopf law. Two complementary †-SCFA's that also form a bialgebra are called strongly complementary.

Strongly Complementary Bases

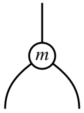
- **[Kissinger et al. 1203.4988]:** Strongly complementary observables in **FHilb** are characterized by Abelian groups.
- Given a finite group G, its multiplication is:



$$G \times G \xrightarrow{m} G$$

Strongly Complementary Bases

- **[Kissinger et al. 1203.4988]:** Strongly complementary observables in **FHilb** are characterized by Abelian groups.
- Given a finite group G, its multiplication is:

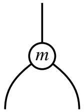


$$G \otimes G \xrightarrow{m} G$$

We linearize this to obtain the group algebra multiplication.

Strongly Complementary Bases

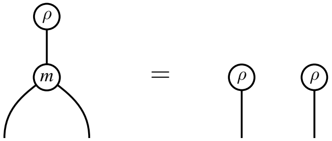
- **[Kissinger et al. 1203.4988]:** Strongly complementary observables in **FHilb** are characterized by Abelian groups.
- Given a finite group G, its multiplication is:



$$G \otimes G \xrightarrow{m} G$$

We linearize this to obtain the group algebra multiplication.

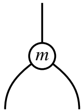
- A *one-dimensional representation* $G \xrightarrow{\rho} \mathbb{C}$ is:



It is copied by the multiplication vertex.

Strongly Complementary Bases

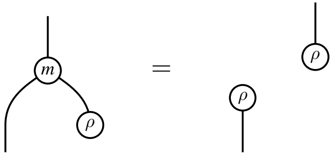
- **[Kissinger et al. 1203.4988]:** Strongly complementary observables in **FHilb** are characterized by Abelian groups.
- Given a finite group G, its multiplication is:



$$G \otimes G \xrightarrow{m} G$$

We linearize this to obtain the group algebra multiplication.

- A *one-dimensional representation* $G \xrightarrow{\rho} \mathbb{C}$ is:



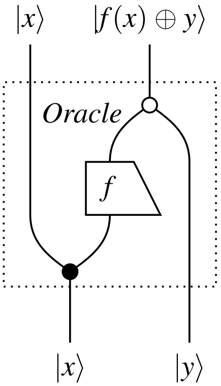
The adjoint $\mathbb{C} \xrightarrow{\rho} G$ is also copied on the lower legs.

Strongly Complementary Bases

- **[Kissinger et al. 1203.4988]:** Strongly complementary observables in **FHilb** are characterized by Abelian groups.
- **[Gogioso & WZ]:** Pairs of strongly complementary observables correspond to Fourier transforms between their bases.*

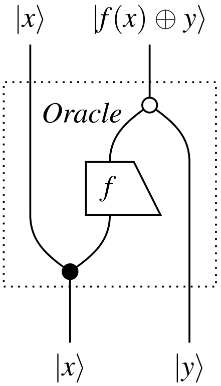
Unitary Oracles

- From these can construct the internal structure of oracles:



Unitary Oracles

- From these can construct the internal structure of oracles:



- **[WZ & Vicary 1406.1278]:** For f to map between bases is a self-conjugate comonoid homomorphism. Oracles with this abstract structure are unitary in general.

Ex 1. The Deutsch-Jozsa Algorithm

- Blackbox function $f : \{0,1\}^N \to \{0,1\}$ is *balanced* when it takes each possible value the same number of times

$$\begin{tikzcd}
11 \arrow[dr] & \\
10 \arrow[dr] & 1 \\
01 \arrow[ur] & 0 \\
00 \arrow[ur] &
\end{tikzcd}$$

Ex 1. The Deutsch-Jozsa Algorithm

- Blackbox function $f : \{0,1\}^N \to \{0,1\}$ is *balanced* when it takes each possible value the same number of times

$$\begin{tikzcd}
11 \arrow[dr] & \\
10 \arrow[dr] & 1 \\
01 \arrow[ur] & 0 \\
00 \arrow[ur] &
\end{tikzcd}$$

Definition (The Deutsch-Jozsa problem)

Given a blackbox function f promised to be either *constant* or *balanced*, identify which.

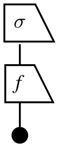
- Classically we require at most $2^{N-1} + 1$ queries of f
- The quantum algorithm only requires a *single* query.

Ex 1. The Deutsch-Jozsa Algorithm

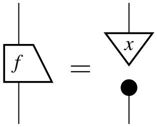
- Blackbox function $f : \{0,1\}^N \to \{0,1\}$ is *balanced* when it takes each possible value the same number of times

$$\begin{tikzcd}
11 \arrow[dr] & \\
10 \arrow[dr] & 1 \\
01 \arrow[ur] & 0 \\
00 \arrow[ur] &
\end{tikzcd}$$

- Let σ be non-trivial irrep. of $\mathbb{Z}_2$ i.e. $\sigma(0) = 1, \sigma(1) = -1$.

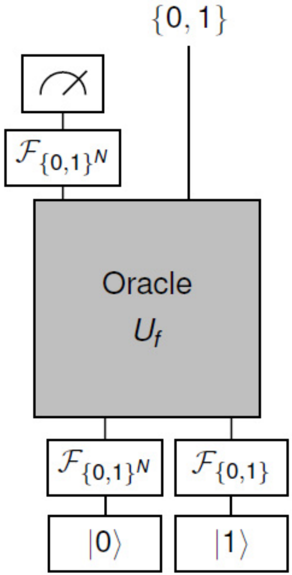
balanced: 

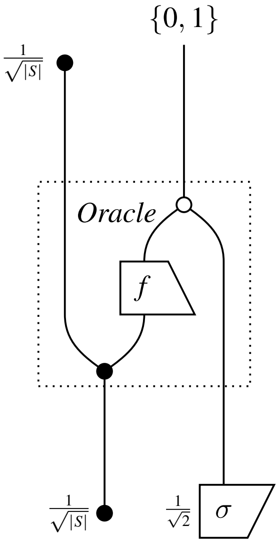
$= 0$

constant: 

Vicary arXiv 1209.3917

Ex 1. The Deutsch-Jozsa Algorithm

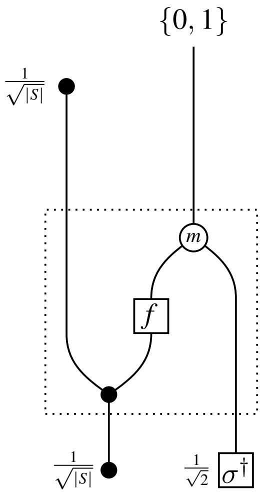




Vicary arXiv 1209.3917

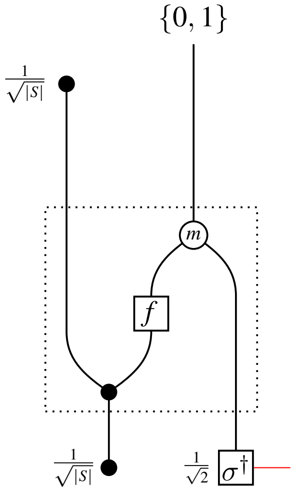
Ex 1. The Deutsch-Jozsa Algorithm

We can use our higher level description to decompose the algorithm:



Ex 1. The Deutsch-Jozsa Algorithm

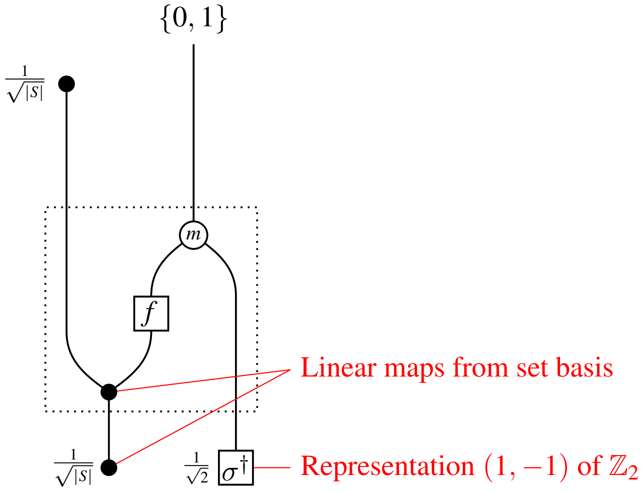
We can use our higher level description to decompose the algorithm:



Representation $(1,-1)$ of $\mathbb{Z}_2$

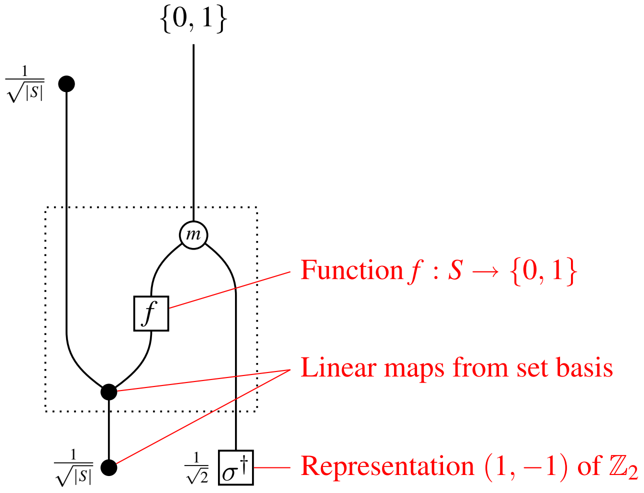
Ex 1. The Deutsch-Jozsa Algorithm

We can use our higher level description to decompose the algorithm:



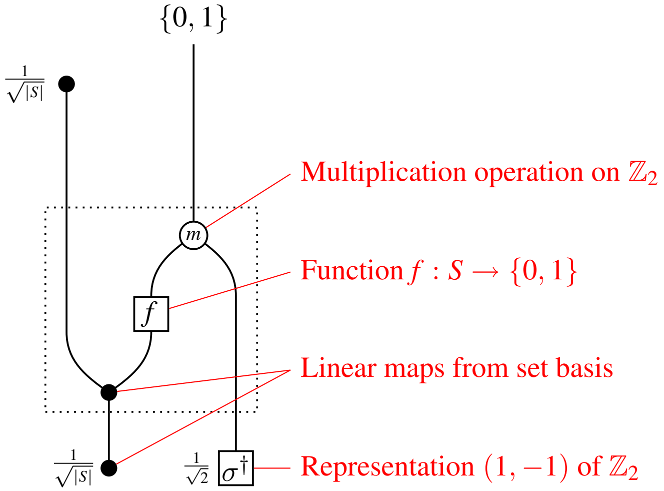
Ex 1. The Deutsch-Jozsa Algorithm

We can use our higher level description to decompose the algorithm:


$\{0,1\}$
$\frac{1}{\sqrt{|S|}}$
m
f
Function $f : S \to \{0,1\}$
Linear maps from set basis
$\frac{1}{\sqrt{|S|}}$
$\frac{1}{\sqrt{2}}$
$\sigma^\dagger$
Representation $(1,-1)$ of $\mathbb{Z}_2$

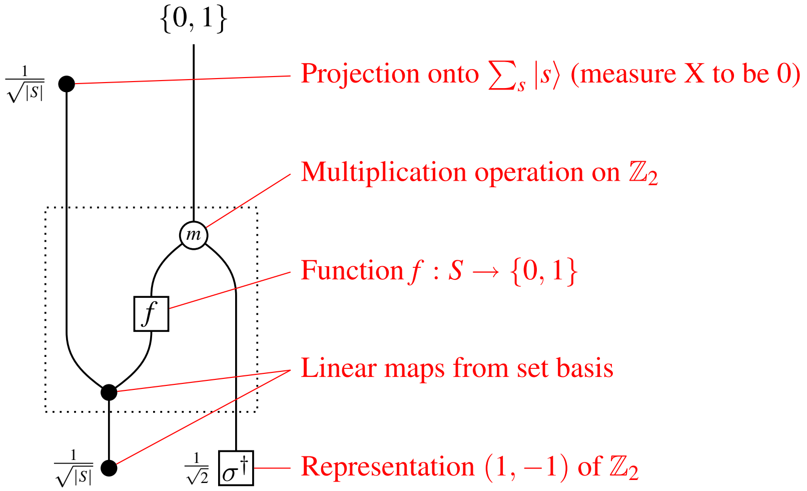
Ex 1. The Deutsch-Jozsa Algorithm

We can use our higher level description to decompose the algorithm:


$\{0,1\}$
$\frac{1}{\sqrt{|S|}}$
m
f
$\frac{1}{\sqrt{|S|}}$
$\frac{1}{\sqrt{2}}$
$\sigma^\dagger$
Multiplication operation on $\mathbb{Z}_2$
Function $f : S \to \{0,1\}$
Linear maps from set basis
Representation $(1,-1)$ of $\mathbb{Z}_2$

Ex 1. The Deutsch-Jozsa Algorithm

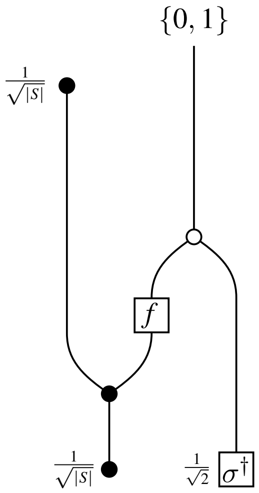
We can use our higher level description to decompose the algorithm:



$\{0, 1\}$
$\frac{1}{\sqrt{|S|}}$
Projection onto $\sum_s |s\rangle$ (measure X to be 0)
Multiplication operation on $\mathbb{Z}_2$
m
Function $f : S \to \{0, 1\}$
f
Linear maps from set basis
$\frac{1}{\sqrt{|S|}}$
$\frac{1}{\sqrt{2}}$
$\sigma^\dagger$
Representation $(1, -1)$ of $\mathbb{Z}_2$

Ex 1. The Deutsch-Jozsa Algorithm

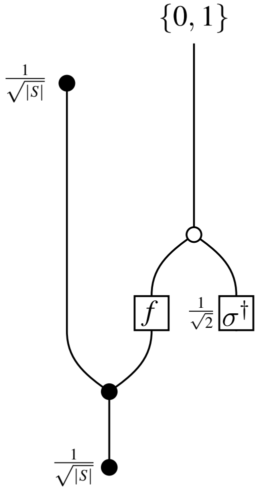
Diagrammatic moves allow us to verify the algorithm in generality:



Vicary arXiv 1209.3917

Ex 1. The Deutsch-Jozsa Algorithm

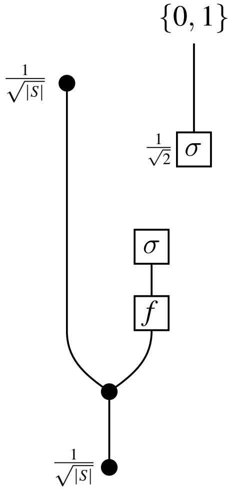
Diagrammatic moves allow us to verify the algorithm in generality:



- Slide up $\sigma^\dagger$

Ex 1. The Deutsch-Jozsa Algorithm

Diagrammatic moves allow us to verify the algorithm in generality:

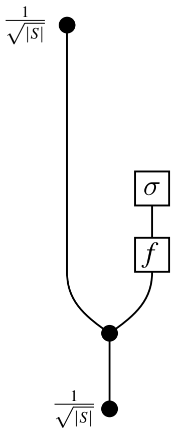


- Slide up $\sigma^\dagger$
- Pull $\sigma^\dagger$ through the whitedot

Vicary arXiv 1209.3917

Ex 1. The Deutsch-Jozsa Algorithm

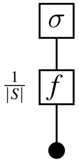
Diagrammatic moves allow us to verify the algorithm in generality:



- Slide up $\sigma^\dagger$
- Pull $\sigma^\dagger$ through the whitedot
- Neglect the right-side system

Ex 1. The Deutsch-Jozsa Algorithm

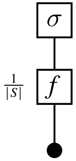
Diagrammatic moves allow us to verify the algorithm in generality:



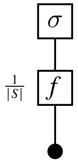
- Slide up $\sigma^\dagger$
- Pull $\sigma^\dagger$ through the whitedot
- Neglect the right-side system
- Topological contraction of blackdot

Ex 1. The Deutsch-Jozsa Algorithm

Gives the amplitude for the input state $\frac{1}{\sqrt{|S|}}\sum_s |s\rangle$ to be in the σ state at measurement.

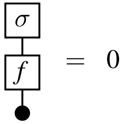


Ex 1. The Deutsch-Jozsa Algorithm



Gives the amplitude for the input state $\frac{1}{\sqrt{|S|}} \sum_s |s\rangle$ to be in the σ state at measurement.

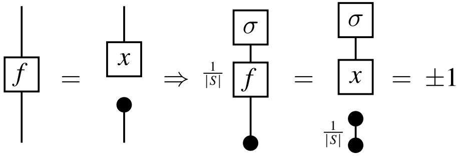
What if f is balanced?



so the system is never measured in σ.

What if f is constant?

Then



So the system is always measured in σ.

Vicary arXiv 1209.3917

Ex 1. Summary for Deutsch-Josza

- Verify: Abstractly verify the algorithm

Ex 1. Summary for Deutsch-Josza

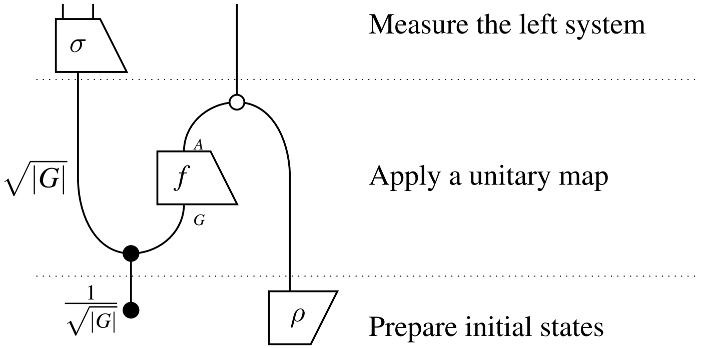
- Verify: Abstractly verify the algorithm
- Generalize:
    - Abstract definition for balanced generalizes **[Høyer Phys. Rev. A 59, 3280 1999]** and **[Batty, Braunstein, Duncan 0412067]**. See **[Vicary 1209.3917]**.

Ex 1. Summary for Deutsch-Josza

- Verify: Abstractly verify the algorithm
- Generalize:
  - Abstract definition for balanced generalizes **[Høyer Phys. Rev. A 59, 3280 1999]** and **[Batty, Braunstein, Duncan 0412067]**. See **[Vicary 1209.3917]**.
  - The algorithm can be executed with complementary rather than strongly complementary observables

Ex 2. The GROUPHOMID Algorithm

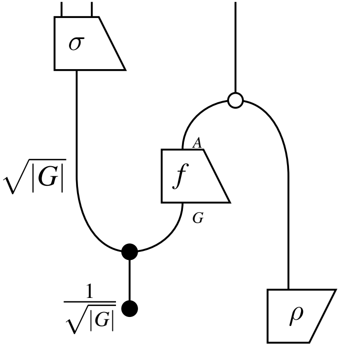
- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.



WZ & Vicary arXiv 1406.1278

Ex 2. The GROUPHOMID Algorithm

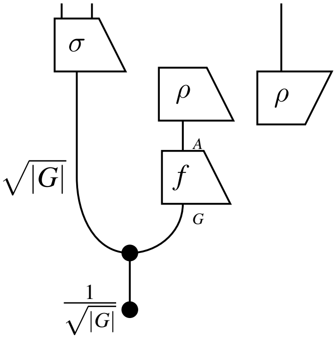
- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.



Ex 2. The GROUPHOMID Algorithm

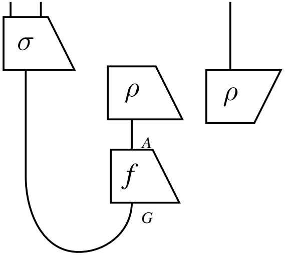
- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.

- Pull ρ through whitedot



Ex 2. The GROUPHOMID Algorithm

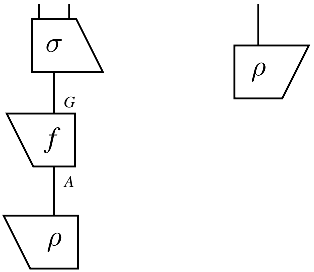
- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.



- Pull ρ through whitedot
- Contract set scalars

Ex 2. The GROUPHOMID Algorithm

- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.

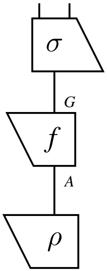


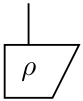
- Pull ρ through whitedot
- Contract set scalars
- Topological equivalence

Ex 2. The GROUPHOMID Algorithm

- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.

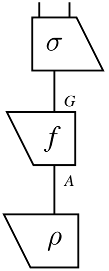
- $\rho \circ f$ is an irrep. of G.

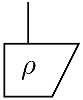




Ex 2. The GROUPHOMID Algorithm

- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.

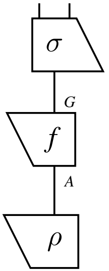


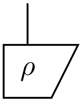


- $\rho \circ f$ is an irrep. of G.
- Choose ρ to be a faithful representation of A.

Ex 2. The GROUPHOMID Algorithm

- Given finite groups G and A where A is abelian, and a blackbox function $f : G \longrightarrow A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.

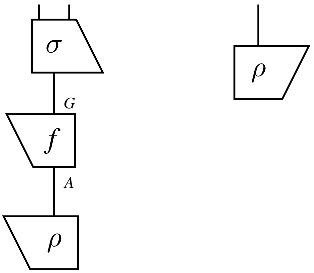




- $\rho \circ f$ is an irrep. of G.
- Choose ρ to be a faithful representation of A.
- Then measuring $\rho \circ f$ identifies f (up to isomorphism)

Ex 2. The GROUPHOMID Algorithm

- Given finite groups G and A where A is abelian, and a blackbox function $f : G \to A$ promised to be a group homomorphism, identify f.
- Case: Let A be a cyclic group $\mathbb{Z}_n$.



- $\rho \circ f$ is an irrep. of G.
- Choose ρ to be a faithful representation of A.
- Then measuring $\rho \circ f$ identifies f (up to isomorphism)
- One-dimensional representations are isomorphic only if they are equal.

Ex 2. The GROUPHOMID Algorithm

The General Case: Homomorphism $f : G \to A$

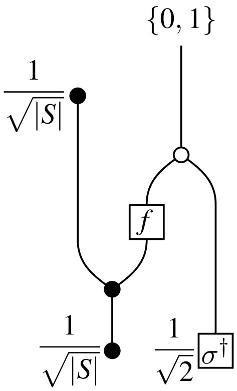
- We generalize with proof by induction via the Structure Theorem. $A = Z_{p_1} \oplus \ldots \oplus Z_{p_k}$
- **[WZ & Vicary 1406.1278]** Given types, the quantum algorithm can identify a group homomorphism in k oracle queries.

Ex 2. The GROUPHOMID Algorithm

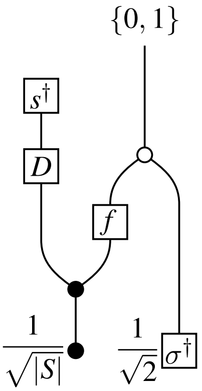
The General Case: Homomorphism $f : G \to A$

- We generalize with proof by induction via the Structure Theorem. $A = Z_{p_1} \oplus \ldots \oplus Z_{p_k}$
- **[WZ & Vicary 1406.1278]** Given types, the quantum algorithm can identify a group homomorphism in k oracle queries.
- Note that the quantum algorithm depends on the structure of A while a classical algorithm will depend on the structure of G.
- **Theorem [WZ]** For large G this algorithm makes a quantum optimal number of queries, while classical algorithms are lower bounded by $\log |G|$.

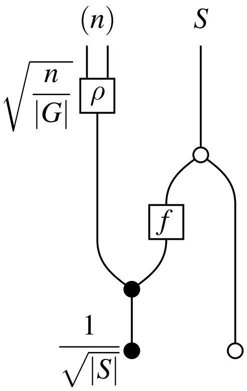
Quantum algorithms: old, generalized and new



Deutsch-Jozsa



Single-shot Grover

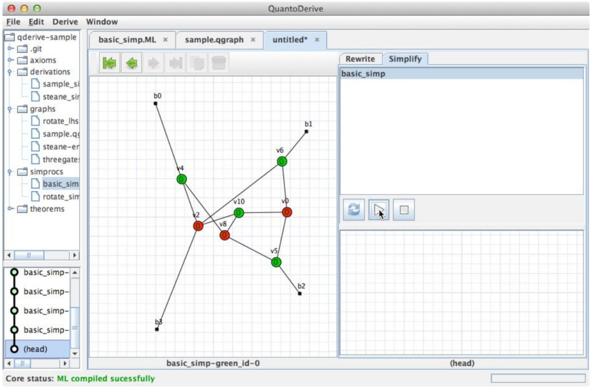


Hidden subgroup

Vicary arXiv 1209.3917   WZ & Vicary arXiv 1406.1278

Other results

- Automated graphical reasoning: `quantomatic.github.io`

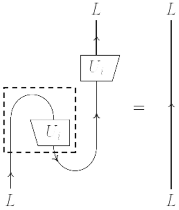


Core status: ML compiled sucessfully

Kissinger arXiv:1203.0202   Dixon et al. arXiv 1007.3794

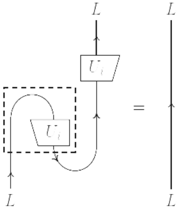
Other results

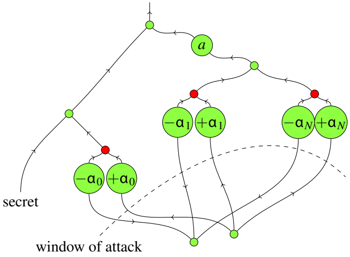
- Automated graphical reasoning: `quantomatic.github.io`
- **[Coecke & Abramsky 0808.1023]** Teleportation



Other results

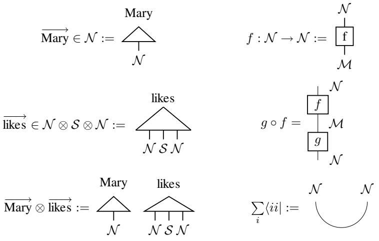
- Automated graphical reasoning: `quantomatic.github.io`
- **[Coecke & Abramsky 0808.1023]** Teleportation
- **[Zamdzhiev 2012, WZ & Gogioso arXiv tmrw]** Quantum Secret Sharing
- **[Cohn-Gordon 2012]** Quantum Bit Commitment





Other results

- Automated graphical reasoning: `quantomatic.github.io`
- **[Coecke & Abramsky 0808.1023]** Teleportation
- **[Zamdzhiev 2012, WZ & Gogioso arXiv tmrw]** Quantum Secret Sharing
- **[Cohn-Gordon 2012]** Quantum Bit Commitment
- Connections to other theories in †-SMC's: **[WZ & Coecke]** DisCo NLP.


$\overrightarrow{\text{Mary}} \in \mathcal{N} :=$
Mary
$\mathcal{N}$
$f : \mathcal{N} \to \mathcal{N} :=$
$\mathcal{N}$
f
$\mathcal{M}$
$\overrightarrow{\text{likes}} \in \mathcal{N} \otimes \mathcal{S} \otimes \mathcal{N} :=$
likes
$\mathcal{N}\,\mathcal{S}\,\mathcal{N}$
$g \circ f =$
$\mathcal{N}$
f
$\mathcal{M}$
g
$\mathcal{N}$
$\overrightarrow{\text{Mary}} \otimes \overrightarrow{\text{likes}} :=$
Mary
likes
$\mathcal{N}$
$\mathcal{N}\,\mathcal{S}\,\mathcal{N}$
$\sum_i \langle ii| :=$
$\mathcal{N}$
$\mathcal{N}$

Coecke et al. 1003.4394 Grefenstette & Sadrzadeh arXiv 1106.4058

Other results

- Automated graphical reasoning: `quantomatic.github.io`
- **[Coecke & Abramsky 0808.1023]** Teleportation
- **[Zamdzhiev 2012, WZ & Gogioso arXiv tmrw]** Quantum Secret Sharing
- **[Cohn-Gordon 2012]** Quantum Bit Commitment
- Connections to other theories in †-SMC's: **[WZ & Coecke]** DisCo NLP.
- **[Kissinger et al. 1203.4988, WZ & Gogioso arXiv tmrw]** Foundations: Mermin Non-locality.
- **[WZ 1503.05857]** Models of quantum algorithms in sets and relations.

Outlook: Use this knowledge of quantum structure to better advantage in quantum programming.

Other results

- Automated graphical reasoning: `quantomatic.github.io`
- **[Coecke & Abramsky 0808.1023]** Teleportation
- **[Zamdzhiev 2012, WZ & Gogioso arXiv tmrw]** Quantum Secret Sharing
- **[Cohn-Gordon 2012]** Quantum Bit Commitment
- Connections to other theories in †-SMC's: **[WZ & Coecke]** DisCo NLP.
- **[Kissinger et al. 1203.4988, WZ & Gogioso arXiv tmrw]** Foundations: Mermin Non-locality.
- **[WZ 1503.05857]** Models of quantum algorithms in sets and relations.

Outlook: Use this knowledge of quantum structure to better advantage in quantum programming.