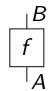
Picturing Quantum Processes II: ZX-calculus and automation

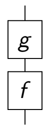
Aleks Kissinger

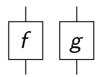
Radboud University

Coalgebraic Methods in Computer Science
Thessalonaki 2018

The setting: symmetric monoidal categories

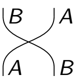
$f : A \to B \quad :=$ 

$g \circ f :=$ 

$f \otimes g :=$ 

$1_A :=$ 

$1_I :=$

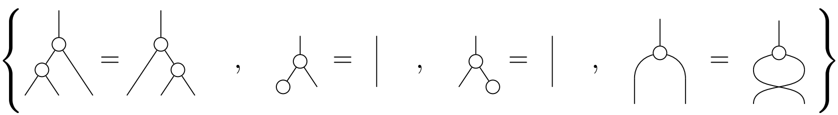
$\sigma_{A,B} :=$ 

Symmetric monoidal theories

- We work with (co)algebraic structures inside an SMC, via *symmetric monoidal theories*
- Symmetric monoidal theories generalise (universal) algebraic theories
- They consist of a set of generators with input/output arities e.g.

$\Sigma :=$ {  $: 2 \to 1$,  $: 0 \to 1$ }

- And a set of *relations*, which are pairs of (formal) compositions in the SMC, respecting arities, e.g.

$E :=$ 

Symmetric monoidal theories

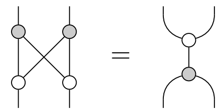
- Relations $(L = R) \in E$ extend to larger diagrams via *substitution*
- In the usual (term-like) picture, we can see this is rewriting *modulo* the SMC axioms:

$$L = R \vdash G = H \quad \iff \quad \exists C_1, C_2, X. \begin{cases} G \overset{smc}{=} C_1 \circ (L \otimes 1_X) \circ C_2 \\ H \overset{smc}{=} C_1 \circ (R \otimes 1_X) \circ C_2 \end{cases}$$

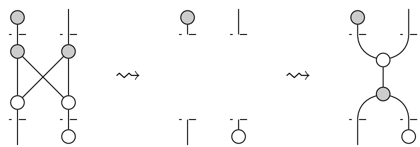
- Rewriting modulo is hard in general
- Simpler to perform substitution *directly on string diagrams*

Equational reasoning with diagram substitution

- For example:



- and be applied as:



Models

- A *model* of (Σ, E) consists of an object $A \in \mathcal{C}$ and morphisms:

$$[\![f]\!] : \underbrace{A \otimes \ldots \otimes A}_{m} \to \underbrace{A \otimes \ldots \otimes A}_{n}$$

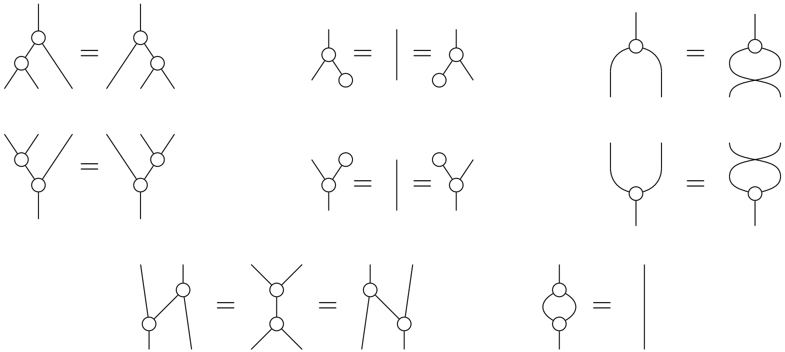
  for all $f : m \to n \in \Sigma$, such that compositions satisfy the equations in E.
- Equivalently, we can define models of (Σ, E) in terms of its associated PROP.
- A PROP is an SMC whose objects are $\mathbb{N}$ ($:=$ (co)arities).
- The *syntactic PROP* $\mathrm{Syn}(\Sigma, E)$ of a theory (Σ, E) has as mophisms string diagrams of Σ-generators, modulo E
- Models are strong monoidal functors $[\![-]\!] : \mathrm{Syn}(\Sigma, E) \to \mathcal{C}$

Semantic PROPs

- Equality in $\mathrm{Syn}(\Sigma, E)$ is undecidable in general
- However, some theories have particularly nice, 'semantic' PROPs with decidable equality.
- e.g. for commutative monoids, $\mathrm{Syn}(\Sigma, E) \simeq (\mathrm{FinSet}, +)$

Example

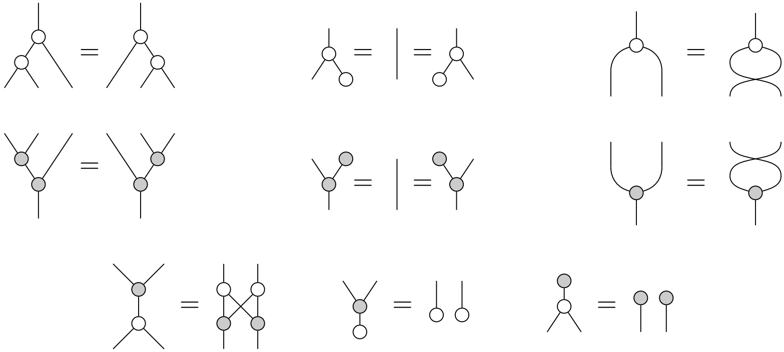
(Special commutative) Frobenius algebras:



n.b. $\mathrm{Syn}(\Sigma, E) \simeq (\mathrm{Csp}(\mathrm{FinSet}), +)$

Example

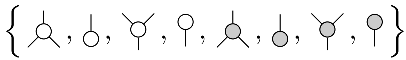
Commutative bialgebras:



n.b. $\mathrm{Syn}(\Sigma, E) \simeq (\mathrm{Mat}[\mathbb{N}], \oplus)$

The system $\mathbb{IB}$

- It is interesting to consider multiple, interacting theories
- $\mathbb{IB}$ (for interacting bialgebras) consists of two **bialgebras** that interact with each other as **Frobenius algebras**
- equivalently, it's two **Frobenius algebras** that interact as **bialgebras**
- Formally, it consists of:

$\Sigma_{\mathbb{IB}} =$ 

such that these are **Frobenius algebras**:

these are **bialgebras**:

and caps/cups coincide:

Interacting bialgebras are linear relations

$$\mathrm{Syn}(\Sigma_{\mathbb{IB}}, E_{\mathbb{IB}}) \cong \mathrm{LinRel}_{\mathbb{Z}_2} \qquad \text{[BSZ'14]}^{1}$$

- $\mathrm{LinRel}_{\mathbb{Z}_2}$ has:
  - **objects:** $\mathbb{N}$
  - **morphisms:** $R : m \to n$ is a subspace:

$$R \subseteq \mathbb{Z}_2^m \times \mathbb{Z}_2^n \cong \mathbb{Z}_2^{m+n}$$

  - **composition** is relation-style. For $R : m \to n$, $S : n \to k$:

$$(u|w) \in S \circ R \quad \iff \quad \exists v \in \mathbb{Z}_2^n.\ (u|v) \in R \ \wedge \ (v|w) \in S$$

  - **tensor** is $\times$ (aka. $\oplus$)

[1] F. Bonchi, P. Sobocinksi, F. Zanasi. *Interacting bialgebras are Frobenius*

Semantic picture

- We can define an interpretation $[\![-]\!] : \mathrm{Syn}(\Sigma_{\mathbb{IB}}, E_{\mathbb{IB}}) \to \mathrm{LinRel}_{\mathbb{Z}_2}$ as:

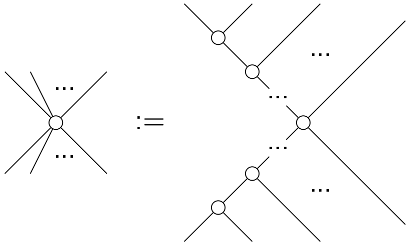
$[\![$  $]\!] = \left\{ \begin{pmatrix} 0 \\ \hline 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ \hline 1 \\ 1 \end{pmatrix} \right\} \subseteq \mathbb{Z}_2^{1+2}$ $\qquad$ $[\![$  $]\!] = \left\{ (0), (1) \right\} \subseteq \mathbb{Z}_2^{1+0}$

$[\![$  $]\!] = \left\{ \begin{pmatrix} 0 \\ 0 \\ \hline 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ \hline 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ \hline 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \hline 0 \end{pmatrix} \right\} \subseteq \mathbb{Z}_2^{2+1}$ $\qquad$ $[\![$  $]\!] = \left\{ (0) \right\} \subseteq \mathbb{Z}_2^{0+1}$

- BSZ showed that $[\![-]\!]$ extends to a PROP iso
- They used the techique of *composing PROPs* via distributive laws

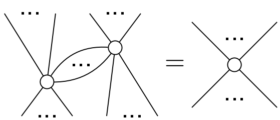
Syntactic picture

- Can also show this syntactically, by string diagram rewriting
- For this, its useful to switch to *unbiased* presentations of Frobenius algebras and bialgebras, via *spiders*:



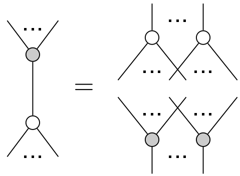
Unbiased Frobenius algebras

All Frobenius equations are subsumed by ‘spider fusion’:

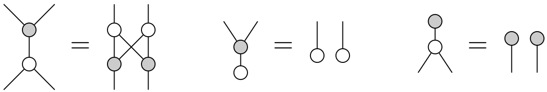


Unbiased bialgebras

All bialgebra laws are subsumed by replacing two connected spiders by a complete biparitte graph:



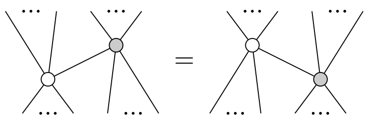
The three basic laws are special cases:



Cups and caps

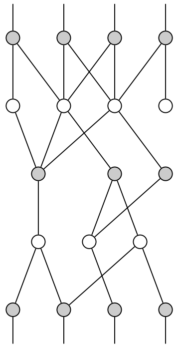
Coincidence of cups and caps:

...can be subsumed by treating string diagrams as undirected, i.e. we can flip wires at will:



A simple rewriting strategy[2]

An arbitrary $\mathbb{IB}$ diagram has many alternating layers of ○ / ●:

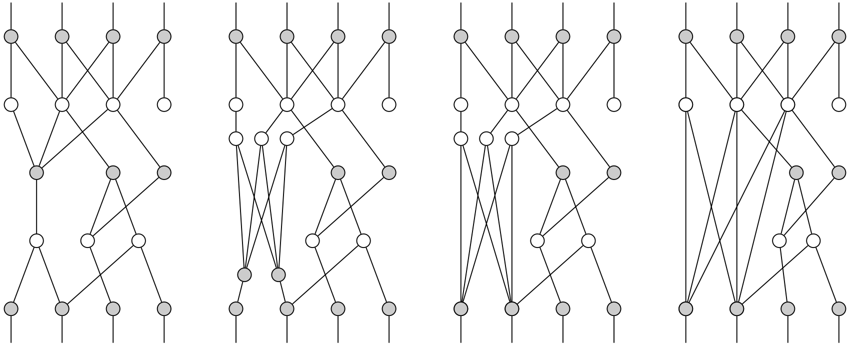


GOAL: make just 3 layers ●—○—●.

[2] F. Bonchi, F. Gadducci, A. Kissinger, P. Sobocinksi, F. Zanasi. *Rewriting with Frobenius*

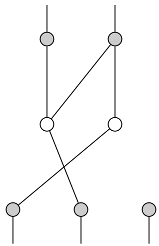
A simple rewriting strategy

STRATEGY: find an *interior* ○-spider, apply generalised bialgebra, then fuse as much as possible.



A simple rewriting strategy

Every iteration removes at least one interior ◯, and doesn't introduce any new ones, so it terminates, with just three layers, e.g.

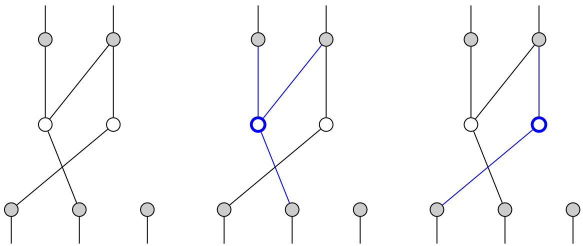


We can read off the subspace from this pseudo-NF:

- ◯-spiders are 'placeholders'
- ○-spiders are 'basis vectors'
- edges represent 1's in the basis vectors at a given place.

Pseudo-normal forms

- Subspaces can be represented as:

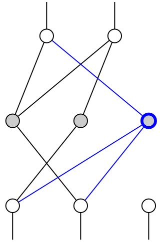


$$\leftrightarrow \quad \left\langle \begin{pmatrix} 0 \\ 1 \\ 0 \\ \hline 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 0 \\ 1 \end{pmatrix} \right\rangle$$

- The 1's indicate where edges appear for each vector.

Pseudo-normal forms

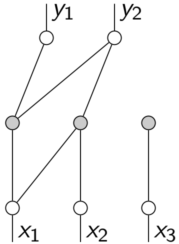
- Not unique! We can always add or remove a vector that is the sum of two other spanning vectors and get the same space:



$$\leftrightarrow \quad \left\langle \begin{pmatrix} 0 \\ 1 \\ 0 \\ \hline 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ \hline 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \\ \hline 1 \\ 0 \end{pmatrix} \right\rangle$$

The dual normal form

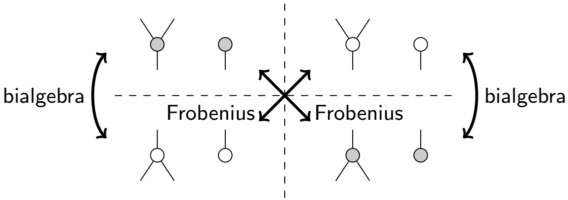
- We can also pass to the dual normal form (grey-white-grey), using the colour-reversed strategy
- This describes the subspace dually, as a system of linear equations:



$$\leftrightarrow \qquad \begin{cases} x_1 = y_1 \oplus y_2 \\ x_1 \oplus x_2 = y_2 \\ x_3 = 0 \end{cases}$$

Automated simplication for $\mathbb{IB}$

We would like to *automated* simplication for $\mathbb{IB}$:

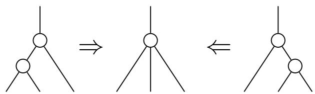


...by turning equations $L = R$ into (directed) rules $L \Rightarrow R$.

...but we have a problem

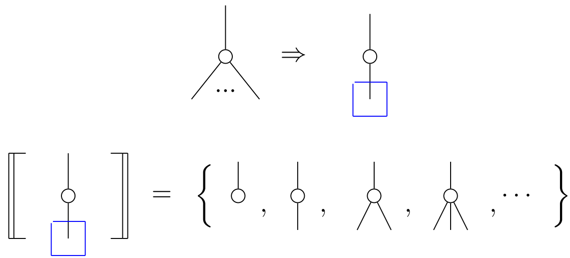
- (Biased) AC rules are not terminating:

- **Solution:** use *unbiased* simplifications, like spider-fusion:

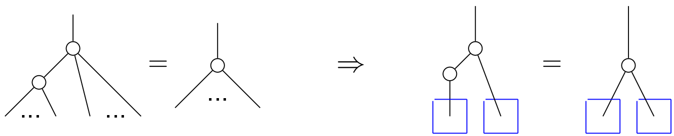


- $\Longrightarrow$ need infinitely many rules, or *rule schemas*

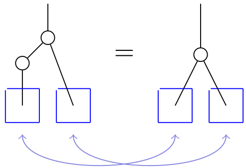
!-boxes: simple diagram schemas

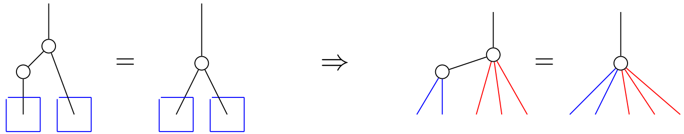


!-boxes: simple diagram rule schemas

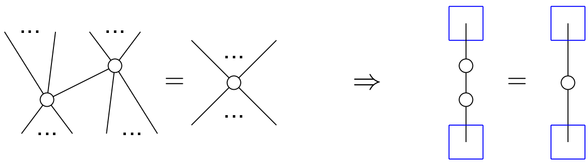


!-boxes

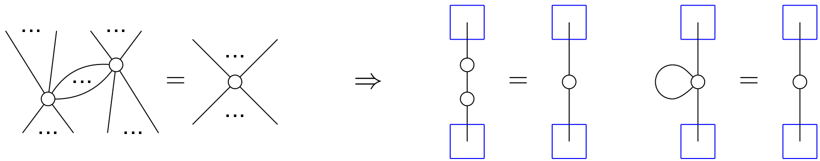




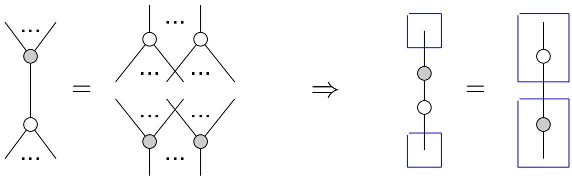
Unbiased rules with !-boxes



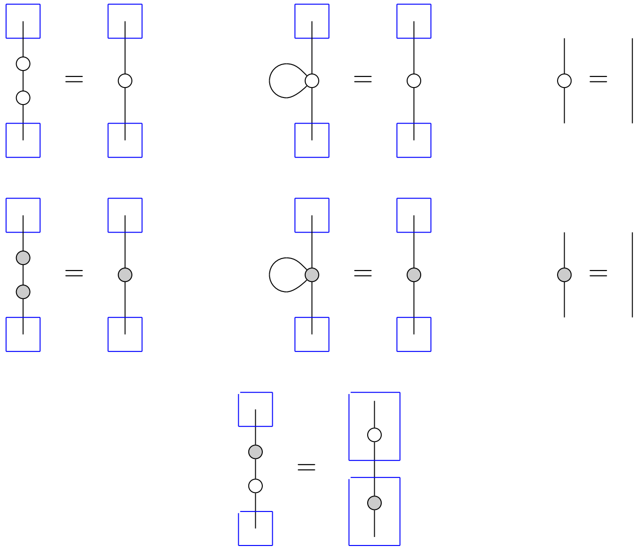
Unbiased rules with !-boxes



Unbiased rules with !-boxes



A !-box presentation of $\mathbb{IB}$

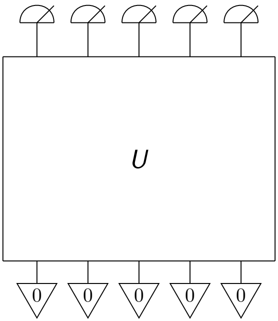


30 rules $\rightsquigarrow$ 7 rules

A !-box presentation of $\mathbb{IB}$

Time to fire up **Quantomatic**.

Quantum computation: the circuit model



- Focus on:

$$U : \underbrace{\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2}_{m} \to \underbrace{\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2}_{n}$$

  where U is a *unitary* linear map ($U^\dagger = U^{-1}$) in $\mathrm{Vect}_{\mathbb{C}}$.
- We can decompose U into smaller unitaries, called *gates*, which we know how to implement on a quantum computer.

The 'quantum trick': unitary oracles

- We have:

$$\underbrace{\mathbb{C}^2 \otimes \ldots \otimes \mathbb{C}^2}_{n} \cong \mathbb{C}^{2^n}$$

- So, we fix a basis of bitstrings called the *computational basis*:

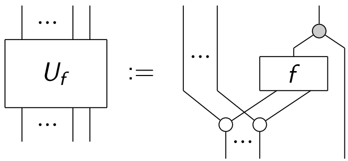
$$\{|00..0\rangle, |0..01\rangle, \ldots, |11..1\rangle\} \subseteq \mathbb{C}^{2^n}$$

- This lets us encode classical functions $F : \{0,1\}^N \to \{0,1\}$ as linear maps:

$$f(|b_1..b_N\rangle) := |F(b_1, .., b_N)\rangle$$

The 'quantum trick': unitary oracles

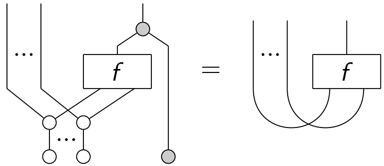
- The $\mathbb{IB}$ generators have an interpretation into $(\mathrm{Vect}_{\mathbb{C}}, \otimes)$
- ...which we can use make the linear map f into a unitary with one weird trick:



which is called the *quantum oracle* of f.

Q: How much does an oracle know?

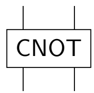
- If we plug in the right state, quite a bit!



- By a good choice of measurements, we can extract **global properties** of f.
- Main trick behind Grover search, Shor's factoring algorithm, etc.

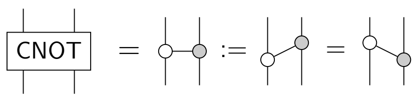
Efficient classical simulation

- The the 'quantum hardness' is in $U : \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$.
- Some U can be represented/computed efficiently, depending on the choice of gates
- Simplest non-trivial 2-qubit gate: CNOT



$$:: \begin{cases} |00\rangle \mapsto |00\rangle \\ |01\rangle \mapsto |01\rangle \\ |10\rangle \mapsto |11\rangle \\ |11\rangle \mapsto |10\rangle \end{cases}$$

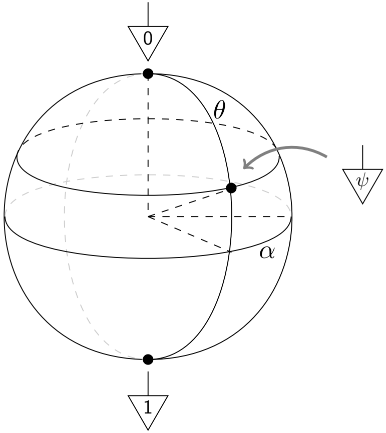
- The $\mathbb{IB}$ generators have a model in $(\mathrm{Vect}_{\mathbb{C}}, \otimes)$ where:



- $\implies$ CNOT circuits are efficiently classically simulable

Adding single-qubit gates

- A single system is a *qubit*, which can be pictured on a sphere:



$$\psi = \begin{pmatrix} c \\ d \end{pmatrix} \propto \begin{pmatrix} \cos(\frac{\theta}{2}) \\ e^{i\alpha}\sin(\frac{\theta}{2}) \end{pmatrix}$$

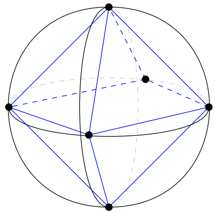
- Unitaries on single qubits $\leftrightarrow$ rotations of the sphere

Adding single-qubit gates

- Adding NOT (180° around X-axis) still gives efficient classical simulation

$$\mathrm{LinRel}_{\mathbb{Z}_2} \rightsquigarrow \mathrm{AffRel}_{\mathbb{Z}_2}$$

- More interesting/quantum: add all rotations preserving this octahedron:



- Gives interesting quantum behaviour (quantum uncertainty/completementarity, non-locality, ...)
- But **still** classically simulable (by Gottesman-Knill theorem)

A complete set of gate identities

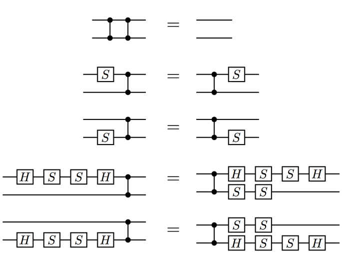
- Octahedron rotations are 2-generated (call generators H and S). Adding to CNOT gives *Clifford circuits*
- The following is a complete set of equations of Clifford circuits:

$$\omega^8 = 1 \tag{C1}$$

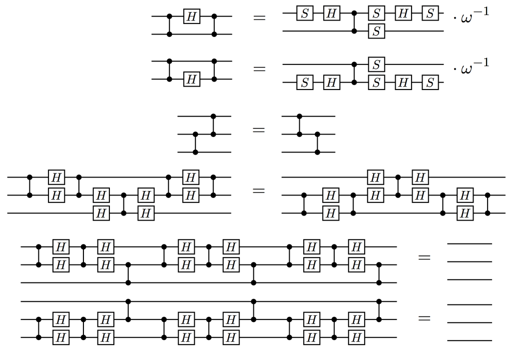
$$H^2 = 1 \tag{C2}$$

$$S^4 = 1 \tag{C3}$$

$$SHSHSH = \omega \tag{C4}$$



(C5) (C6) (C7) (C8) (C9)



(C10) (C11) (C12) (C13) (C14) (C15)

(Selinger 2013)

As an equational theory

- The good:
  - complete for Clifford circuits:

$$[\![C_1]\!] = [\![C_2]\!] \quad \Longrightarrow \quad C_1 =_E C_2$$

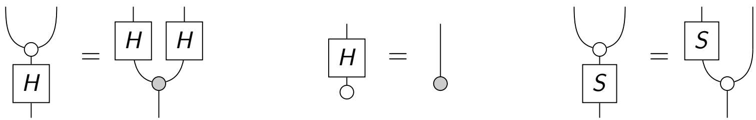
  - unique normal forms
  - relatively compact (3 generators, 15 rules)
- The bad:
  - rules are large, and don't carry any intuition or algebraic structure
  - rewrite strategy is complicated (17 derived gates, 100 derived rules)
- The ugly:
  - proof of completeness is *extremely* complicated ($>$ 100 pages long! though mostly machine-generated)
- Can we do better by extending $\mathbb{IB}$?

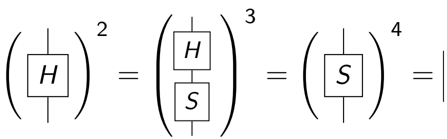
ZX-calculus, presentation 1

Generators:

$$\Sigma_{\mathbb{IB}} \;+\; \left\{ \boxed{S}, \boxed{H} \right\}$$

Equations:

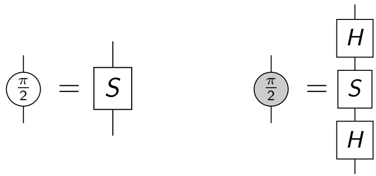


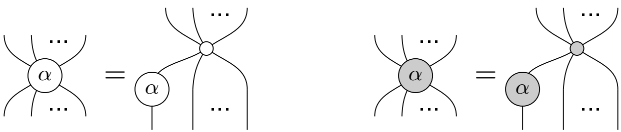


ZX-calculus, unbiased presentation

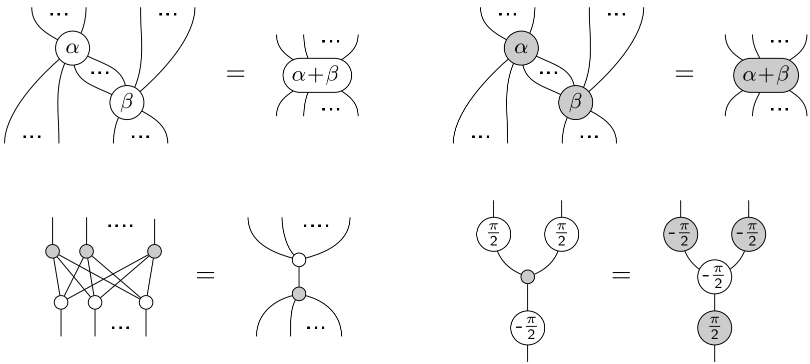
Generators:

These are related to the other generators by:





ZX-calculus, presentation 2



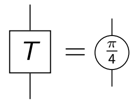
Completeness

Theorem (Backens'10)

The ZX-calculus is complete for Clifford quantum computation.

T gates and universality

- Adding one more generator:



  gives us (approximately) everything.
- For any unitary U, we can find U' built with our generators such that for any $\epsilon > 0$, we have:

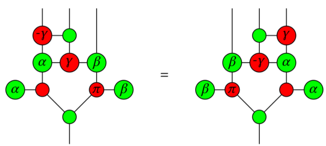
$$U \overset{\epsilon}{=} U'$$

Completeness, take 2

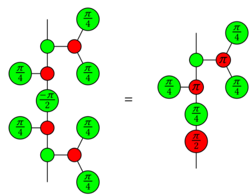
Theorem (JPV'17[3])

The ZX-calculus is complete for Clifford+T quantum computation.

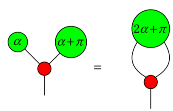
The rules are ZX + 3 more:



(C)



(BW)



(SUP)

[3] Jeandel, Perdrix, and Vilmart. *A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics*

...and even more completeness

	Calculus	Family	Num. Rules
Backens'10	ZX	Clifford	4
Backens'14	ZX	1-qubit Clifford+T	5
Hadzihasanavic'15	ZW	$\mathbb{Z}$-matrices	19
JPV'17	Y	CNOT + $Y(\frac{\pi}{2})$	11
JPV'17	ZX+	Clifford+T	12
Wang & Ng'17	ZX+	ALL	32
JPV'18	ZX+	ALL	13
Wang & Ng'18	ZX+	2-qubit Clifford+T	8
AK & Backens'18	ZH	ALL	11

TODO NOW:

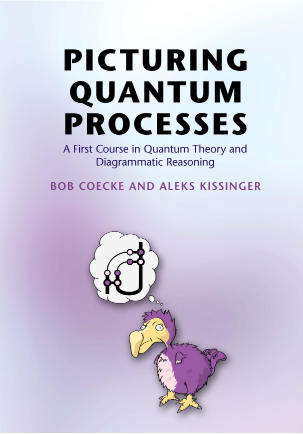
Completeness theorems $\Rightarrow$ (efficent) simplication

Application of techniques beyond QT[4]

[4] Signal flow diagrams, classical circuits, Petri nets, ...

Thanks for your attention

http://quantomatic.github.io



http://cambridge.org/pqp