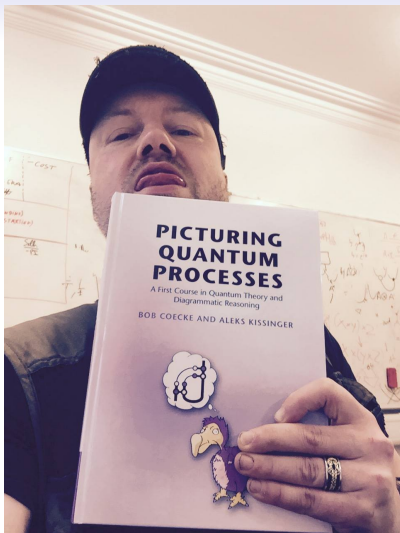


# Picturing Quantum Processes

**Aleks Kissinger** and Bob Coecke

**Radboud University** and Oxford University

ESLLI Toulouse 2017



[www.cambridge.org/pqj](http://www.cambridge.org/pqj)

20% discount @ CUP with code: **COECKE2017**

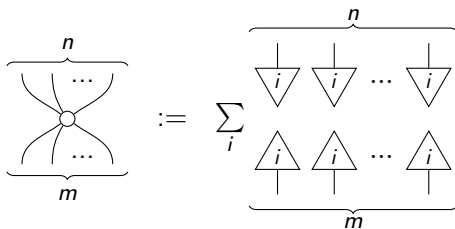
# Chapters 11-13:

## Quantum Foundations, Computation, and Resources

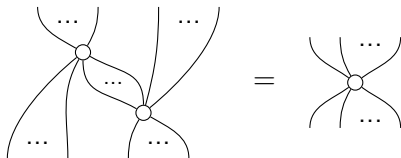
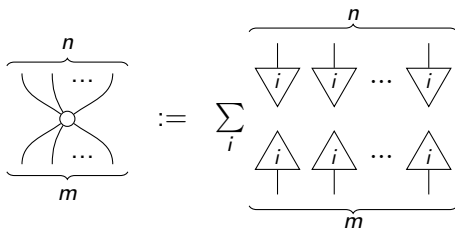
*Mermin once summarized a popular attitude towards quantum theory as “Shut up and calculate”. We suggest a different slogan: “Shut up and contemplate”!*

— Lucien Hardy and Rob Spekkens, 2010.

# Spiders



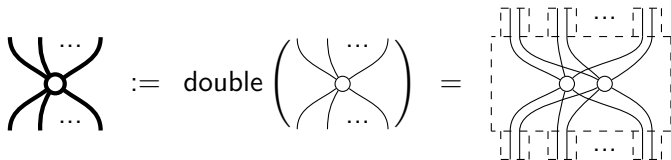
# Spiders



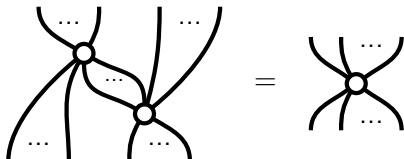
(Classical) spiders are linear maps:



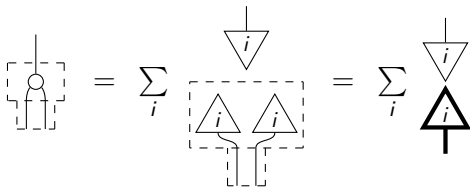
...so we can double them to get *quantum spiders*:



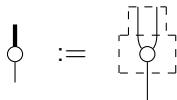
...which also fuse together:



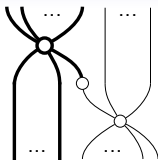
Measuring is a spider:



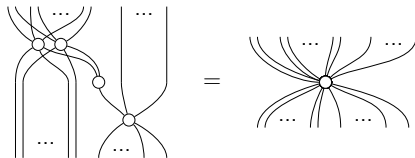
...and so is encoding:



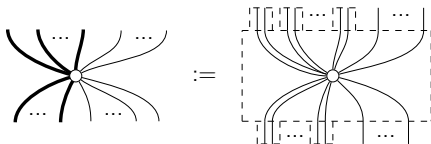
They connect classical to quantum:



...giving something new:

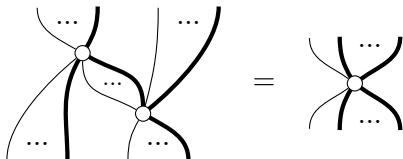


*Bastard spiders!*

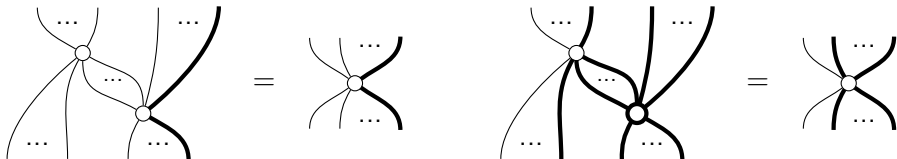




Bastard spiders fuse together:



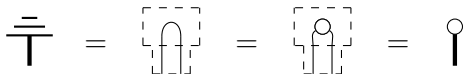
They also absorb other kinds of spider:



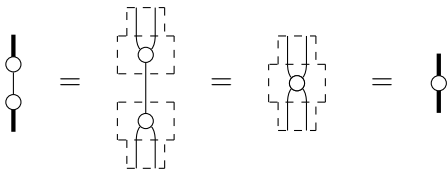
## Useful bastards

Bastard spiders arise naturally in the interaction between classical and quantum data, e.g.

- Discarding:

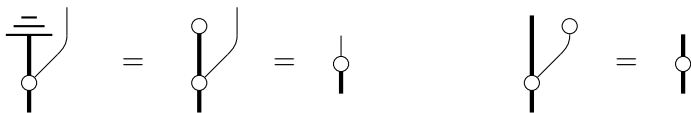
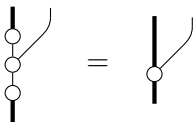


- Decoherence:

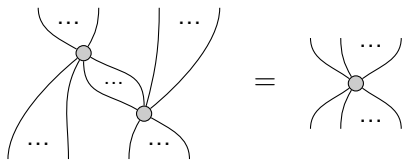
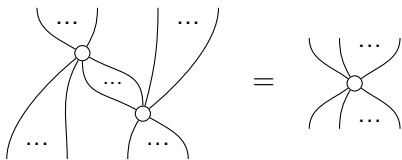
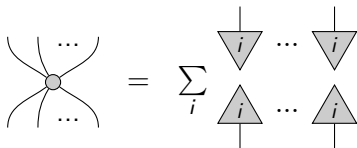
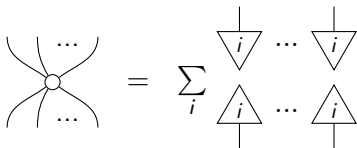


## Useful bastards (cont'd)

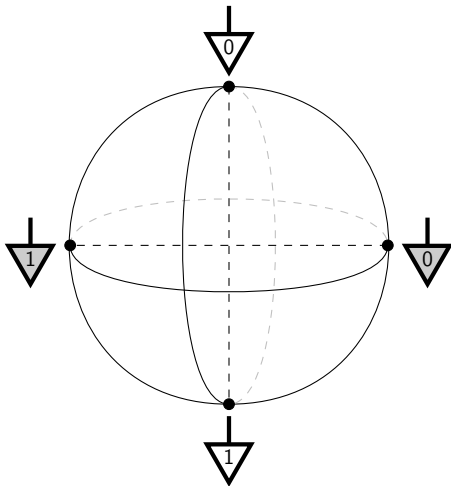
- Non-demolition measurement:



# Multicoloured spiders



# Complementary bases



# Complementarity

**Theorem.** Two bases are complementary (a.k.a. mutually unbiased):

$$\forall i, j : \begin{array}{c} \triangle j \\ | \\ \triangle i \end{array} = \frac{1}{D}$$

if and only if:

$$\begin{array}{c} \bullet \\ \circlearrowleft \\ \circ \\ | \end{array} = \frac{1}{D} \begin{array}{c} | \\ \bullet \\ \circ \\ | \end{array}$$

# Complementarity

**Theorem.** Two bases are complementary (a.k.a. mutually unbiased):

$$\forall i, j : \begin{array}{c} \triangle j \\ | \\ \triangle i \end{array} = \frac{1}{D}$$

if and only if:

$$\begin{array}{c} \bullet \\ | \\ \circ \\ | \end{array} = \frac{1}{D} \begin{array}{c} \bullet \\ | \\ \circ \\ | \end{array}$$

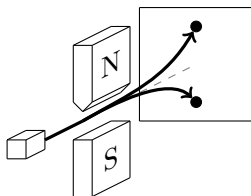
# Complementarity

$$\begin{array}{c} | \\ \bullet \\ \text{---} \\ \circ \\ | \end{array} = \frac{1}{D} \begin{array}{c} | \\ \bullet \\ | \\ \circ \\ | \end{array}$$

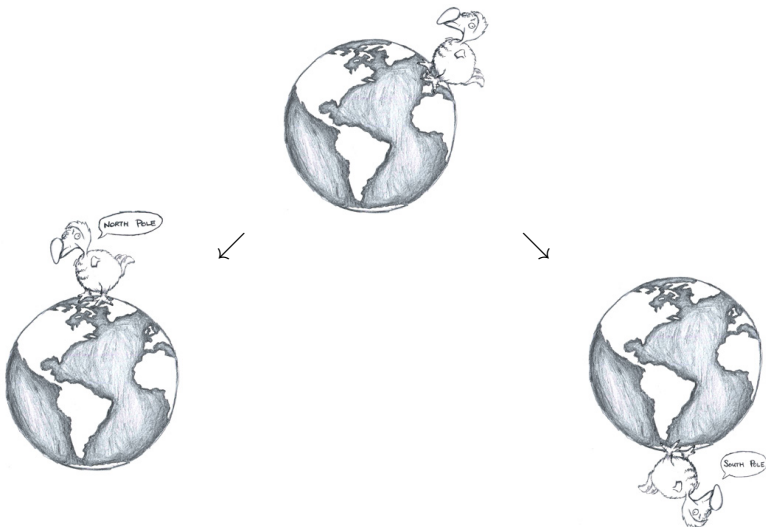
(encode in  $\circ$ ) THEN (measure in  $\bullet$ ) = (no data flow)



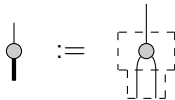
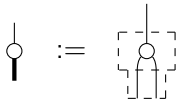
# Stern-Gerlach



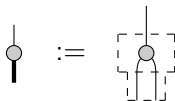
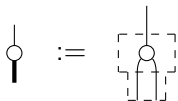
# Stern-Gerlach



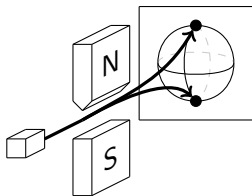
# Stern-Gerlach



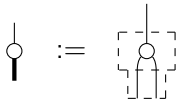
# Stern-Gerlach



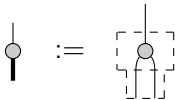
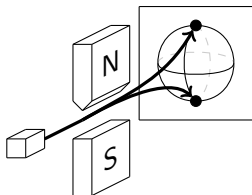
Z-measurement



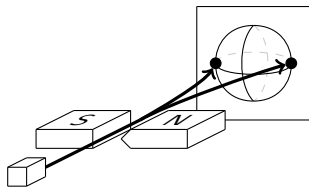
# Stern-Gerlach



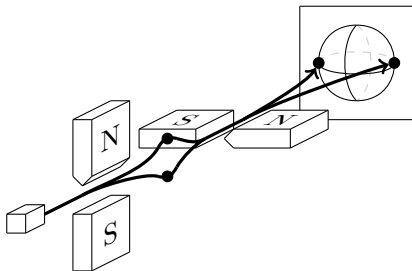
Z-measurement



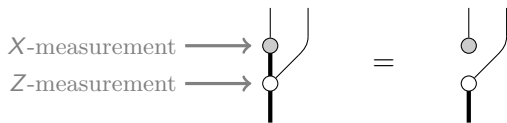
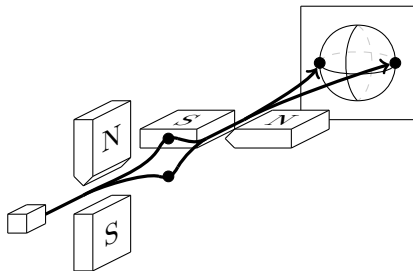
X-measurement



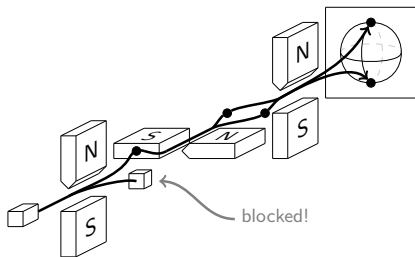
# Stern-Gerlach



# Stern-Gerlach

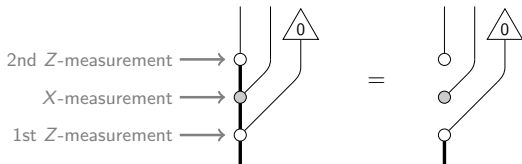
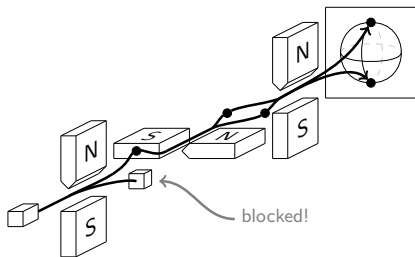


# Stern-Gerlach

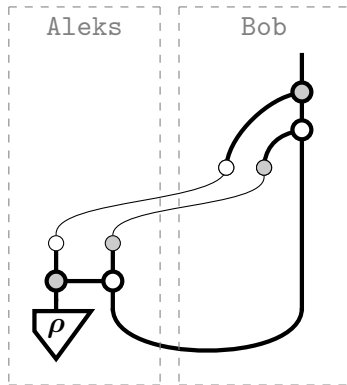




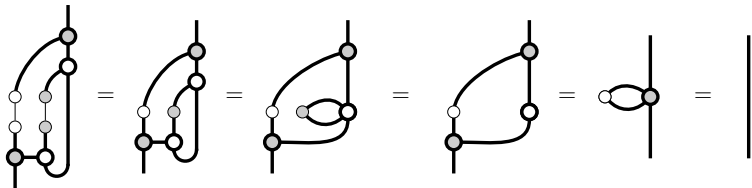
# Stern-Gerlach



# Teleportation with spiders



# Teleportation with spiders



# Power of the Graphical Language

We now have a fairly powerful language, it is natural to ask:

# Power of the Graphical Language

We now have a fairly powerful language, it is natural to ask:

**Q:** Is it *universal*?

Can we express any map in terms of spiders?

# Power of the Graphical Language

We now have a fairly powerful language, it is natural to ask:

**Q:** Is it *universal*?

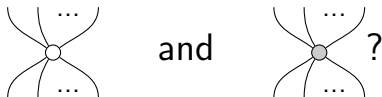
Can we express any map in terms of spiders?

**Q:** Is it *complete*?

Can we prove every equation between maps  
using some set of spider-equations?

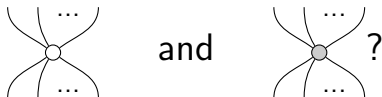
# Universality

**Q:** Can we write any linear map as a diagram of:



# Universality

**Q:** Can we write any linear map as a diagram of:

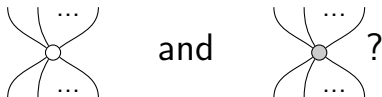


**A:** Clearly not! e.g.  $\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$



# Universality

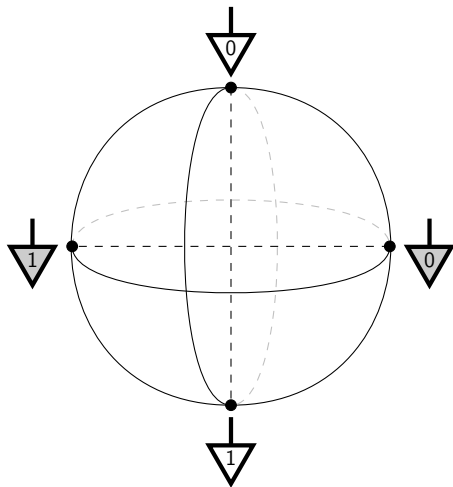
**Q:** Can we write any linear map as a diagram of:



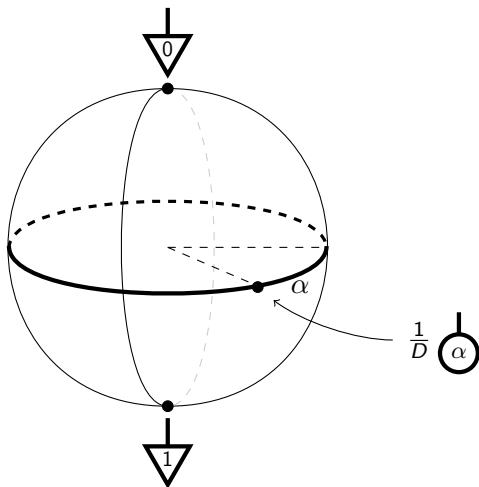
**A:** Clearly not! e.g.  $\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

**Q2:** How much more do we need?

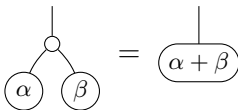
# Phases



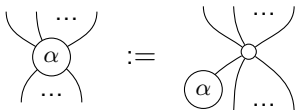
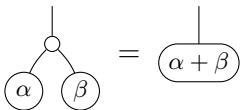
# Phases



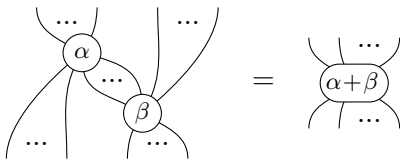
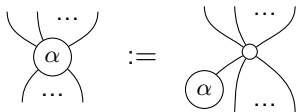
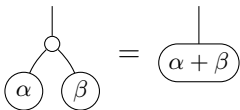
# Phase spiders



# Phase spiders



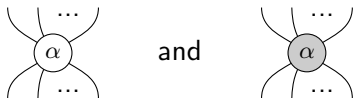
# Phase spiders



# Universality

# Universality

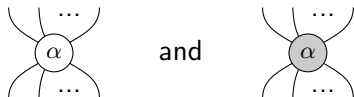
**Theorem.** Any linear map with 2D inputs and outputs can be expressed as a diagram of:



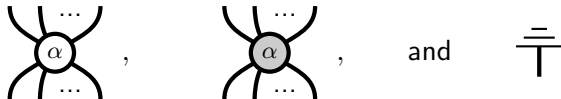


# Universality

**Theorem.** Any linear map with 2D inputs and outputs can be expressed as a diagram of:

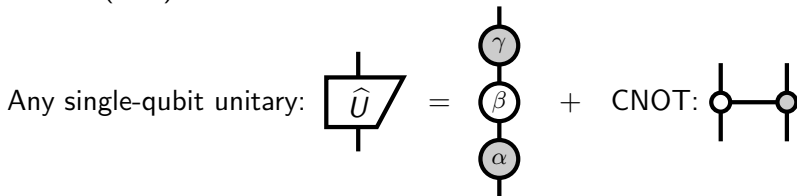


**Corollary.** Any **quantum map** from qubits to qubits can be expressed as a diagram of:

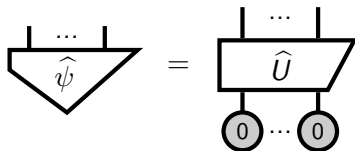


# Universality

**Proof.** (idea)



...gives any unitary, which gives any state:

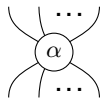


...which gives any map by process-state duality.

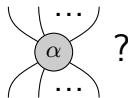
# Completeness

# Completeness

**Q.** Can we find a *complete* set of equations to describe the behaviour of:

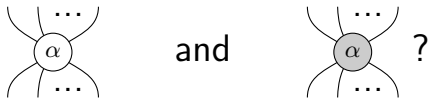


and



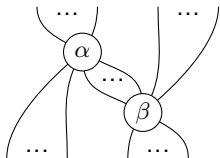
# Completeness

**Q.** Can we find a *complete* set of equations to describe the behaviour of:

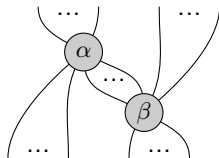


**A.** Yes! The *ZX-calculus*.

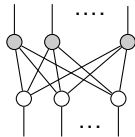
# ZX-calculus



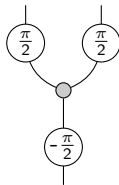
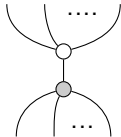
=



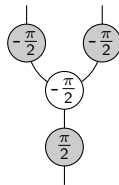
=



=



=



# ZX-calculus

**Theorem.** The ZX-calculus is complete for *Clifford diagrams*, i.e. diagrams where  $\alpha \in \{0, \pi, \pm\frac{\pi}{2}\}$ .

# ZX-calculus

**Theorem.** The ZX-calculus is complete for *Clifford diagrams*, i.e. diagrams where  $\alpha \in \{0, \pi, \pm \frac{\pi}{2}\}$ .

**Theorem.** (Hot off the press! <sup>1,2</sup>) A bigger version of the ZX-calculus is complete for *all diagrams*.

---

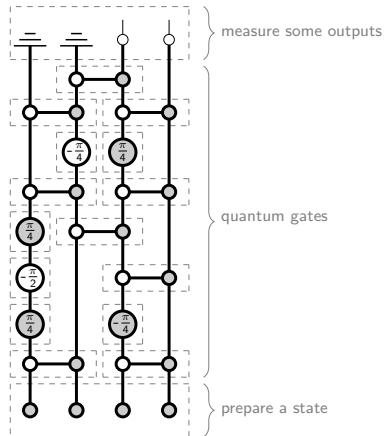
<sup>1</sup>Jeandel, Perdrix, Vilmart. 31 May, 2017. [arXiv:1705.11151](https://arxiv.org/abs/1705.11151)

<sup>2</sup>Wang & Ng. 29 June, 2017. [arXiv:1706.09877](https://arxiv.org/abs/1706.09877)



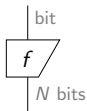
# Application: Quantum Computing

The quantum *circuit model*:



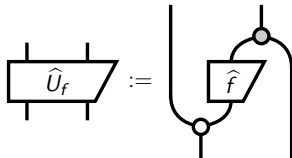
# Quantum algorithms

*Classical computation*

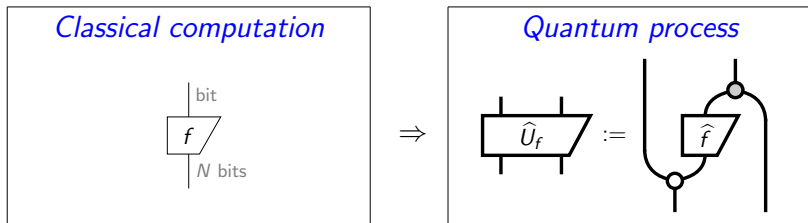


$\Rightarrow$

*Quantum process*



# Quantum algorithms



$\Rightarrow$  **Deutsch-Jozsa, Bernstein-Vazirani, quantum search, and hidden subgroup** (e.g. factoring) algorithms.

# Automation

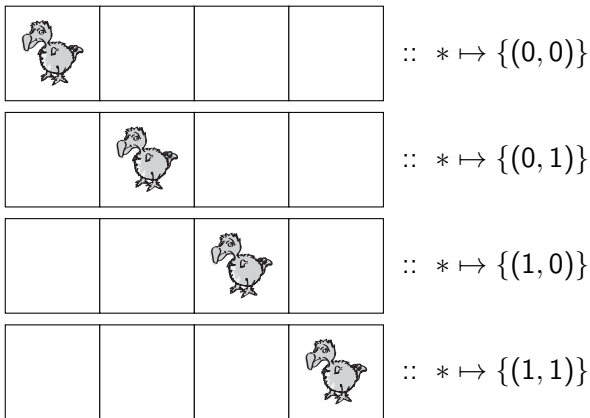
Quantomatic:

The screenshot displays the Quantomatic software interface. The main window is titled "QuantoDerive" and shows a quantum circuit diagram with nodes labeled v3, v4, v10, v11, v12, v13, v14, v15, v16, v17, v18, v19, v20 and b10 through b16. The diagram is divided into two panels: "dir-hopf-1" on the left and "(head)" on the right. The right panel shows a simplified version of the circuit with nodes labeled v5, v4, v11, v10, v14, v13, v12, v11, v10, v9, v8, v7, v6, v5, v4, v3, v2, v1, v0 and b10 through b16. The interface includes a menu bar (File, Edit, Derive, Window, Export), a file explorer on the left, and a code editor on the right. The code editor shows the following rules:

```
dir-rules/mk-cnot (0/0)
dir-rules/red-id (1/1)
dir-rules/red-sp-dir (1/1)
dir-rules/red-sp-dir2 (0/0)
dir-rules/red-sp-split (1/2)
dir-rules/rg-dir (0/0)
dir-rules/r-ir (1/1)
```

The status bar at the bottom indicates "Core status: OK".

# Spekkens' toy model



# Spekkens' toy model

$$Z :: (a, b) \mapsto (a, b \oplus 1)$$

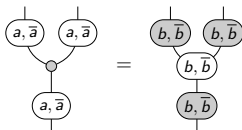
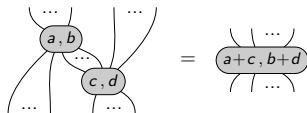
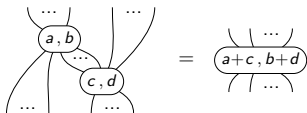
$$Z :: (a, b) \mapsto (a, a \oplus b)$$

$$H :: (a, b) \mapsto (b, a)$$

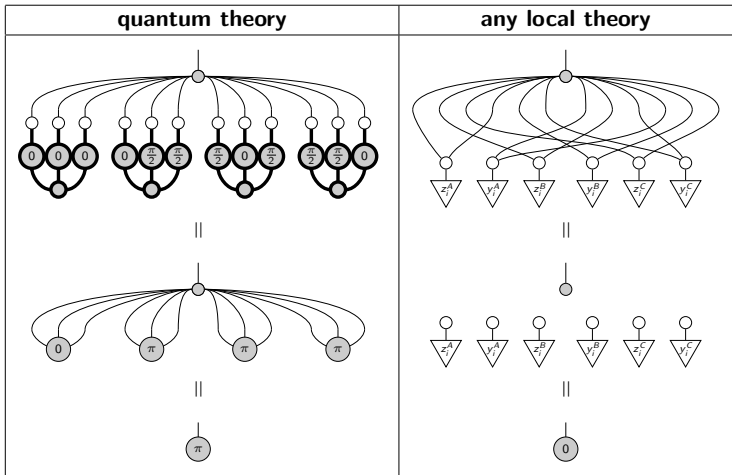
$$CZ :: ((a, b), (c, d)) \mapsto ((a, b \oplus c), (c, a \oplus d))$$

MEASURE( $a, b$ ) := reveal  $a$ , randomize  $b$

# Spekkens' toy model



# GHZ/Mermin non-locality



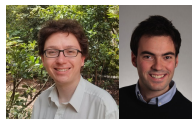


# Thanks!



Coecke, Abramsky, Backens, Duncan, Edwards, Gogioso, Hadzihasanovic, Heunen, Lal, Dixon, Merry, Pavlovic, Perdrix, Quick, Selinger, Zamdzhiev, ...

Special thanks  $\Rightarrow$



Jamie Vicary and David Reutter - qubit.zone simulators

[www.cambridge.org/pqp](http://www.cambridge.org/pqp)

[quantomatic.github.io](http://quantomatic.github.io)