*Vy*ZX

Formal Verification of a Graphical Language

Adrian Lehmann Benjamin Caldwell Bhakti Shah Robert Rand

Department of Computer Science
University of Chicago
Presented at USCS LSD Seminar x UChicago PLRG (October 20, 2023)

The ZX Calculus...

... is a graphical language for reasoning about quantum systems

ZX diagrams are open graphs consisting of green "Z" or red "X" "spiders" and "connections" between them

The Z and X spider

The ZX Calculus...

... is a graphical language for reasoning about quantum systems

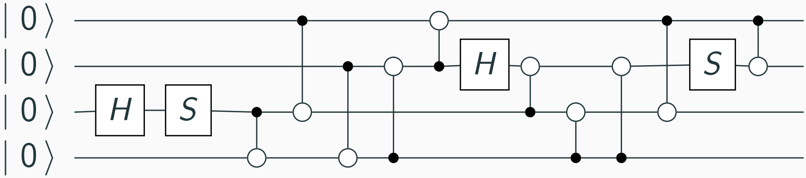
ZX diagrams are open graphs consisting of green "Z" or red "X" "spiders" and "connections" between them

The Z and X spider

Used for compilation, simulation, error correction & more

Key benefit: Diagrammatic rewrites complete & more comprehensible than circuits or matrices

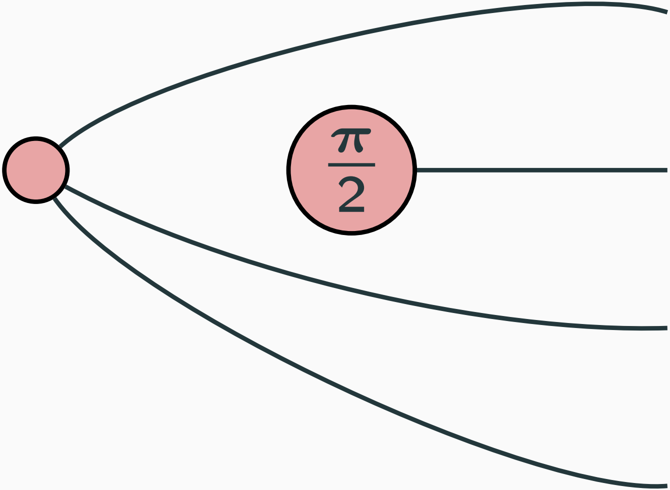
Example: Entanglement (Van de Wetering)



Example: Entanglement (Van de Wetering)

$$[1 \ 0 \ 0 \ 0 \ i \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ i]^\top$$

Example: Entanglement (Van de Wetering)



Qubits!

- Quantum computers use quantum bits or qubits.

Qubits!

- Quantum computers use quantum bits or qubits.
- A qubit is represented in bra-ket notation.
- $|0\rangle$ represents the "ket 0" state ($(1, 0)^T$).
- $|1\rangle$ represents the "ket 1" state ($(0, 1)^T$).

Qubits!

- Quantum computers use quantum bits or qubits.
- A qubit is represented in bra-ket notation.
- $|0\rangle$ represents the "ket 0" state $((1,0)^T)$.
- $|1\rangle$ represents the "ket 1" state $((0,1)^T)$.
- n qubit states can be expressed as $|a_1 a_2 \dots a_n\rangle$ which means $|a_1\rangle \otimes |a_2\rangle \otimes \dots |a_n\rangle$

Qubits!

- Quantum computers use quantum bits or qubits.
- A qubit is represented in bra-ket notation.
- $|0\rangle$ represents the "ket 0" state ($(1, 0)^T$).
- $|1\rangle$ represents the "ket 1" state ($(0, 1)^T$).
- n qubit states can be expressed as $|a_1 a_2 \dots a_n\rangle$ which means $|a_1\rangle \otimes |a_2\rangle \otimes \dots |a_n\rangle$
- $|+\rangle, |-\rangle$ represents the X basis states (transform with hadamard (H))

Quantum Qubit States

- A qubit can be in a superposition of states.

Quantum Qubit States

- A qubit can be in a superposition of states.
- For example, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}$.

Quantum Qubit States

- A qubit can be in a superposition of states.
- For example, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}$.
- The bra notation $\langle\psi|$ is equivalent to $|\psi\rangle^T$.

Quantum Qubit States

- A qubit can be in a superposition of states.
- For example, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \alpha, \beta \in \mathbb{C}$.
- The bra notation $\langle\psi|$ is equivalent to $|\psi\rangle^T$.
- When measuring a qubit, it collapses to either $|0\rangle$ or $|1\rangle$.
- The probability of measuring $|0\rangle$ is $|\alpha|^2$.
- The probability of measuring $|1\rangle$ is $|\beta|^2$.
- The sum of probabilities is always 1: $|\alpha|^2 + |\beta|^2 = 1$.

Quantum Circuit Model

- Quantum operations are represented as gates in a circuit model.
- Each gate acts on qubits, changing their states.
- Common gates include Hadamard (H), Pauli-X (X), Pauli-Y (Y), and Pauli-Z (Z).
- Quantum circuits are read from left to right, and gates are applied in sequence.

Quantum Circuit Model

- Quantum operations are represented as gates in a circuit model.
- Each gate acts on qubits, changing their states.
- Common gates include Hadamard (H), Pauli-X (X), Pauli-Y (Y), and Pauli-Z (Z).
- Quantum circuits are read from left to right, and gates are applied in sequence.

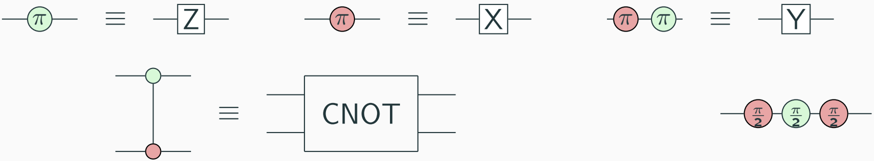
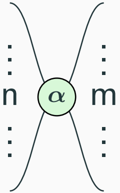
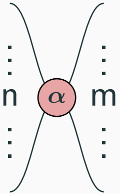


Diagram Semantics



$$= \quad \overset{n}{|0\rangle\cdots|0\rangle} \mapsto \overset{m}{|0\rangle\cdots|0\rangle}$$

$$\overset{n}{|1\rangle\cdots|1\rangle} \mapsto e^{i\alpha}\overset{m}{|1\rangle\cdots|1\rangle}$$



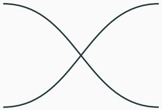
$$= \quad \overset{n}{|-\rangle\cdots|-\rangle} \mapsto \overset{m}{|-\rangle\cdots|-\rangle}$$

$$\overset{n}{|+\rangle\cdots|+\rangle} \mapsto e^{i\alpha}\overset{m}{|+\rangle\cdots|+\rangle}$$

Diagram Semantics

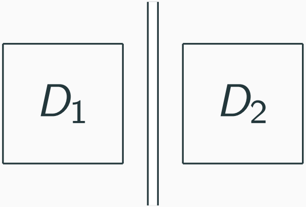
$$\mapsto \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\mapsto \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$$

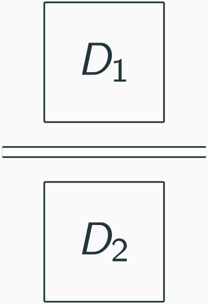


$$\mapsto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Diagram Semantics

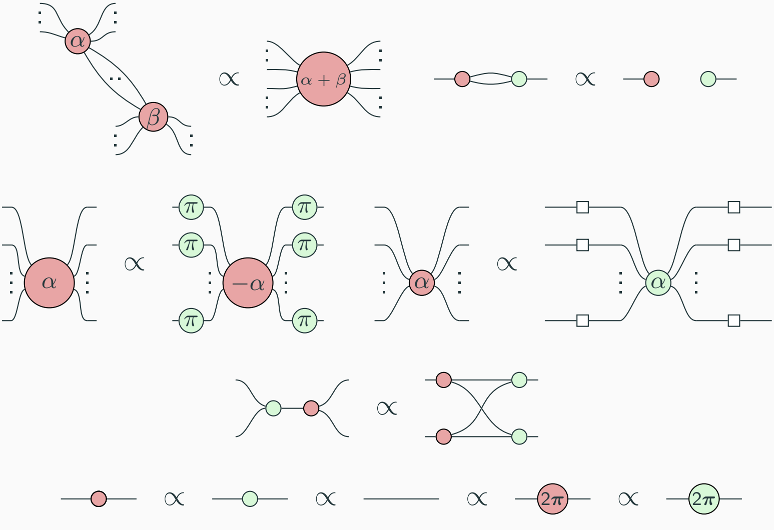


$$\mapsto \quad [D_2] \times [D_1]$$



$$\mapsto \quad [D_1] \otimes [D_2]$$

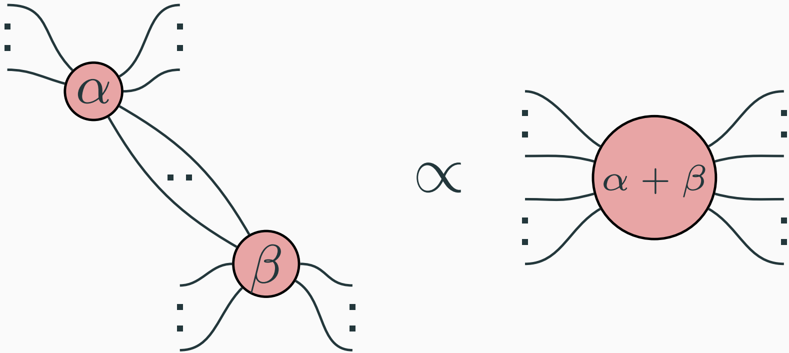
Rewriting Diagrams



Spider Fusion, The Hopf Rule, the Bi-pi Rule (Pi-Copy), Bi-hadamard Rule, the Bialgebra rule, and the identity rule

Spider Fusion

Spider fusion allows us to merge same colored spiders as long as they have one connection, adding their rotations together and connecting any inputs or outputs from the initial two spiders to the final spider.

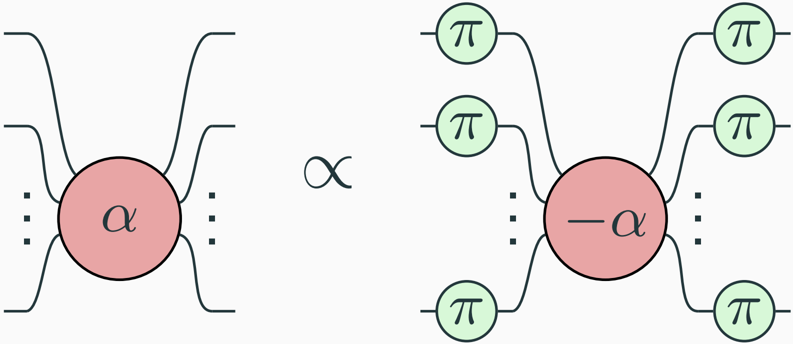


Hopf Rule

The hopf rule allows us to disconnect connections between opposite color spiders that come in pairs of two. A consequence of this with spider fusion is that *n* connections between opposite color spiders can always be considered to be equivalent to *n* mod 2 connections between those spiders.

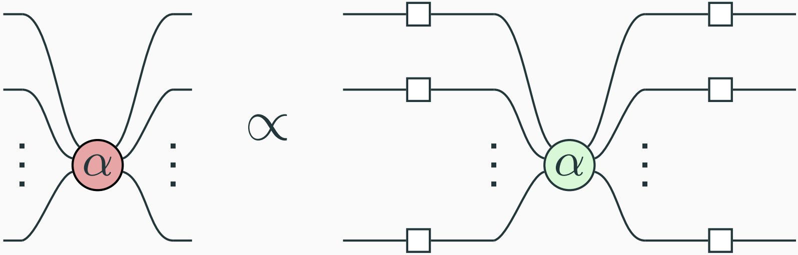
Bi-Pi Rule

The Bi-Pi rule allows us to add spiders of opposite colors with rotation π to every input and output of a spider and flip the phase.



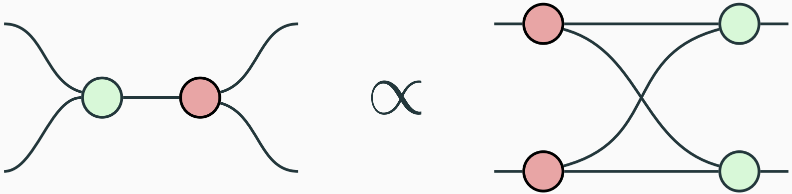
Bi-Hadamard Rule

The Bi-Hadamard rule allows us to add H-boxes to every input and output to flip the color of the spider within



Bialgebra rule

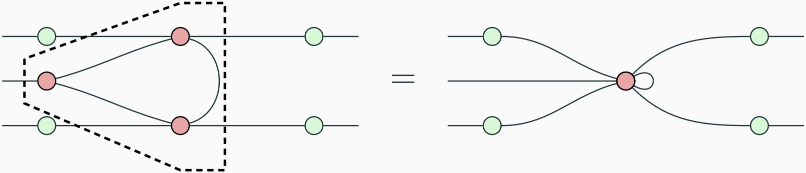
The bialgebra rule is unique in that it is one of the few rules that can introduce or remove swaps.



Identity Removal

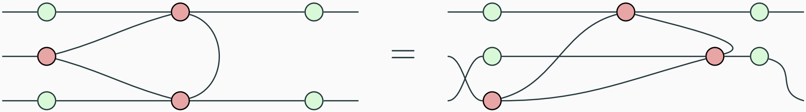
The identity removal rules allow us to remove spiders with $k2\pi$ rotations in general.

Fusion can be used between the three connected X spiders here to simplify our diagram.

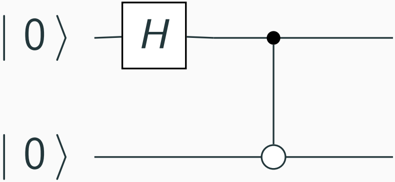


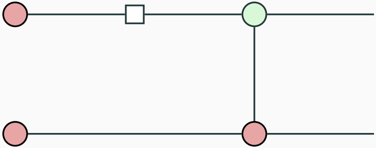
Only Connectivity Matters

We can freely move spiders around, as long as their connections and in/outputs remain the same

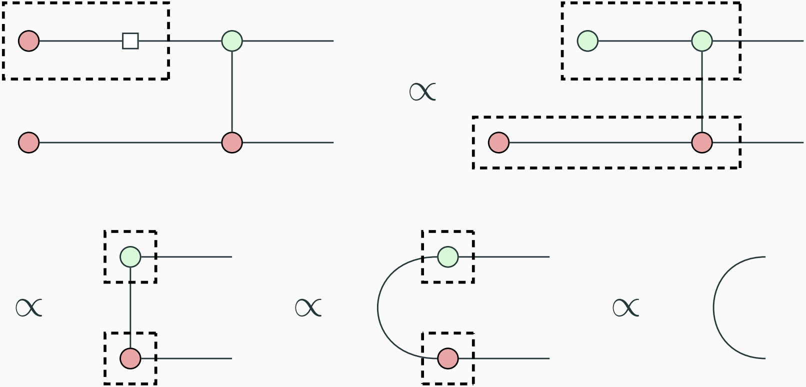


Example: Preparing a Bell Pair

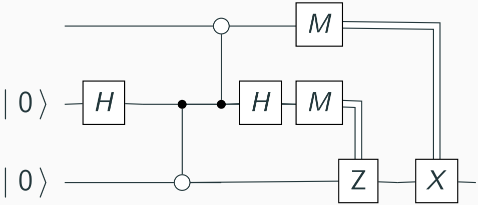


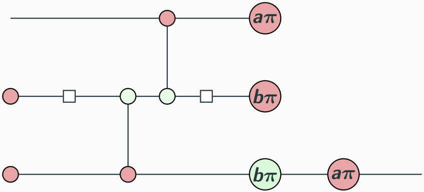


Example: Preparing a Bell Pair

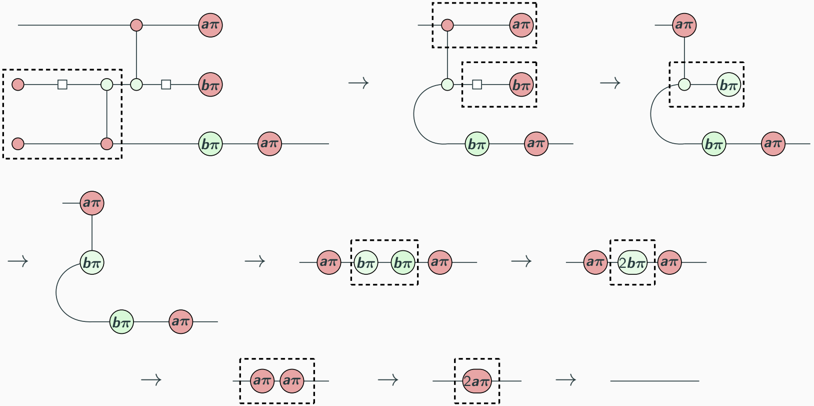


Example: Teleportation (Van de Wetering)





Example: Teleportation (Van de Wetering)



Reasoning Tools

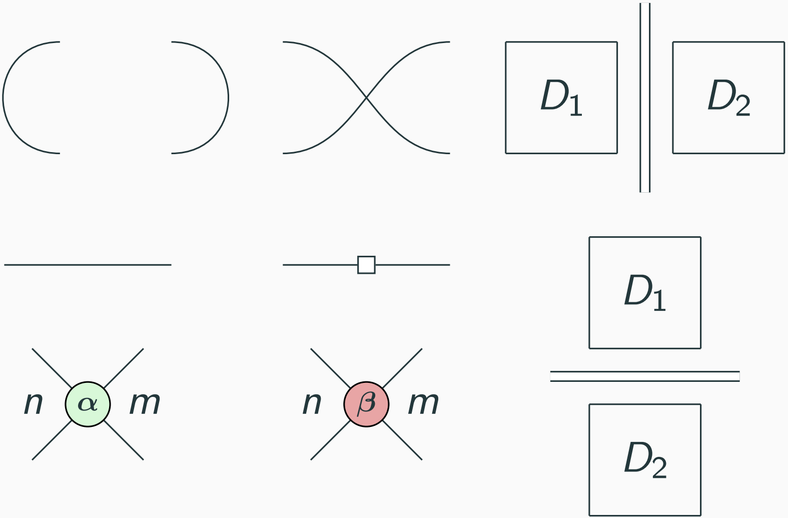
Unverified / Fully axiomatic:

- Quantomatic (https://quantomatic.github.io/)
- ZX Calculator (zx.cduck.me)
- Chyp

"Verify" by property testing:

- PyZX (https://github.com/Quantomatic/pyzx)

ZX Diagrams as string diagrams



Inductive ZX Diagrams

To define our ZX diagrams, we take these string diagram constructions and add Z and X spiders.

$$\frac{\mathsf{in}\ \mathsf{out} : \mathbb{N} \qquad \alpha : \mathbb{R}}{\texttt{Z\_Spider}\ \mathsf{in}\ \mathsf{out}\ \alpha : \texttt{ZX}\ \mathsf{in}\ \mathsf{out}} \qquad \frac{\mathsf{in}\ \mathsf{out} : \mathbb{N} \qquad \alpha : \mathbb{R}}{\texttt{X\_Spider}\ \mathsf{in}\ \mathsf{out}\ \alpha : \texttt{ZX}\ \mathsf{in}\ \mathsf{out}}$$

$$\frac{}{\texttt{Cap} : \texttt{ZX}\ 0\ 2} \qquad \frac{}{\texttt{Cup} : \texttt{ZX}\ 2\ 0} \qquad \frac{}{\texttt{Swap} : \texttt{ZX}\ 2\ 2} \qquad \frac{}{\texttt{Empty} : \texttt{ZX}\ 0\ 0}$$

$$\frac{\mathsf{zx1} : \texttt{ZX}\ \mathsf{in}\ \mathsf{mid} \qquad \mathsf{zx2} : \texttt{ZX}\ \mathsf{mid}\ \mathsf{out}}{\texttt{Compose}\ \mathsf{zx1}\ \mathsf{zx2} : \texttt{ZX}\ \mathsf{in}\ \mathsf{out}} \qquad \frac{}{\texttt{Wire} : \texttt{ZX}\ 1\ 1}$$

$$\frac{\mathsf{zx1} : \texttt{ZX}\ \mathsf{in1}\ \mathsf{out1} \qquad \mathsf{zx2} : \texttt{ZX}\ \mathsf{in2}\ \mathsf{out2}}{\texttt{Stack}\ \mathsf{zx1}\ \mathsf{zx2} : \texttt{ZX}\ (\mathsf{in1} + \mathsf{in2})\ (\mathsf{out1} + \mathsf{out2})} \qquad \frac{}{\texttt{Box} : \texttt{ZX}\ 1\ 1}$$

Semantics

To verify transformations on diagrams, we introduce a system of semantics. Our semantics system will rely on QuantumLib.

$$\texttt{Z\_Spider n m } \alpha \quad \mapsto \quad \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & e^{i\alpha} \end{bmatrix}$$

$$\texttt{X\_Spider n m } \alpha \quad \mapsto \quad H^{\otimes m} \times \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & e^{i\alpha} \end{bmatrix} \times H^{\otimes n}$$

More Semantics

$$
\begin{aligned}
\texttt{Cap} &\mapsto \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix} \\
\texttt{Cup} &\mapsto \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}^\top \\
\texttt{Swap} &\mapsto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
\texttt{Empty} &\mapsto \begin{bmatrix} 1 \end{bmatrix} \\
\texttt{Wire} &\mapsto I_{2\times 2} \\
\texttt{Box} &\mapsto H \\
\texttt{Compose zx1 zx2} &\mapsto \texttt{semantics(zx2)} \times \texttt{semantics(zx1)} \\
\texttt{Stack zx1 zx2} &\mapsto \texttt{semantics(zx1)} \otimes \texttt{semantics(zx2)}
\end{aligned}
$$

Proportionality

Equivalence in ZX is up to constant factor

We define `proportionality` and use symbol $\propto$:

$$\exists c \neq 0 : \texttt{semantics}(zx1) = c * \texttt{semantics}(zx2) \implies zx1 \propto zx2$$

Allows Coq's rewriting capabilities in our proofs about diagrams

Why semantics?

- Smaller TCB
- Interoperability

Why semantics?

- Smaller TCB
- Interoperability

    We can ingest quantum circuits using sqir

    Circuit structure is very different

    Convert circuit components

    Prove equivalence through ground truth

Why semantics?

- Smaller TCB
- Interoperability
    - We can ingest quantum circuits using sqir
    - Circuit structure is very different
    - Convert circuit components
    - Prove equivalence through ground truth
- *Vy*ZX can calculate results

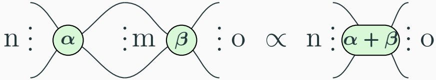
Three Proof Strategies

1. Proof through semantics

Three Proof Strategies

1. Proof through semantics

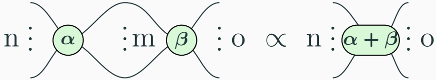
2. Inductive proof



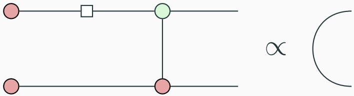
Three Proof Strategies

1. Proof through semantics

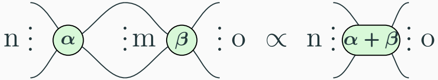
2. Inductive proof



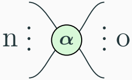
3. Diagrammatic proof



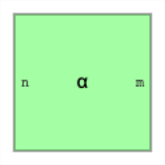
2. Inductive proof



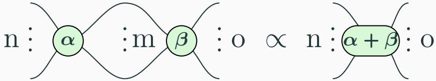
Other representation



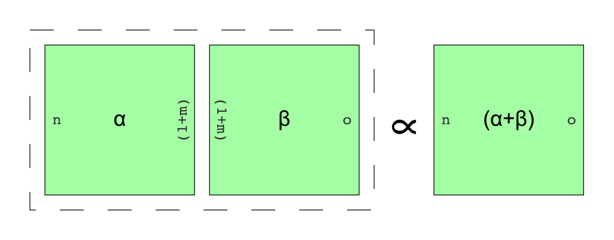
$$\widehat{=}$$



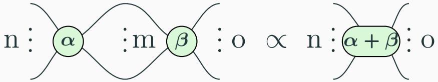
2. Inductive proof: Absolute Fusion



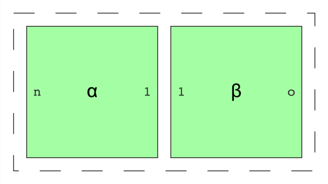
The lemma:



Let's induct over m

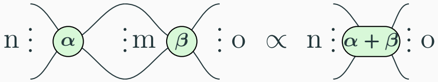


Base case:

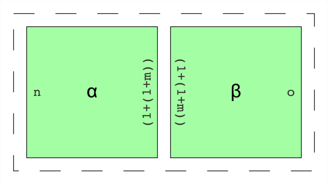


Solve with matrix semantics!

2. Inductive proof: Absolute Fusion

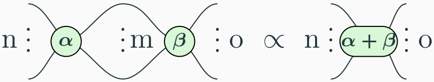


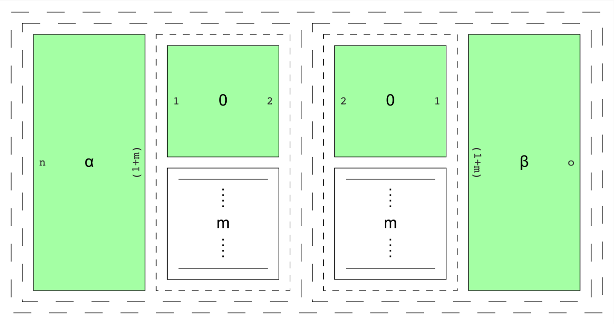
Inductive step:



Let's split out nodes to reduce size

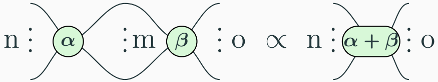
2. Inductive proof: Absolute Fusion



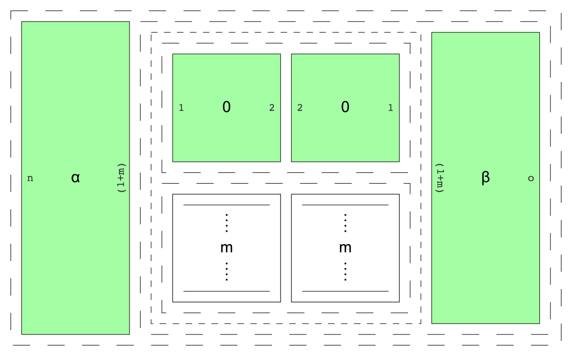


Idea: Fuse the small spiders. Problem: Association

2. Inductive proof: Absolute Fusion

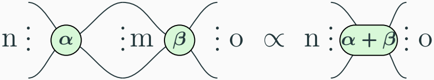


Reassociated diagram

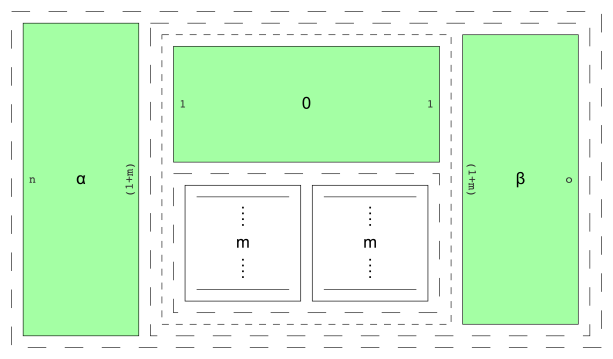


Let's fuse the small spiders

2. Inductive proof: Absolute Fusion



Upper spiders are fused



Apply identity rule, remove wires

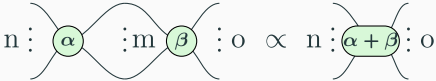
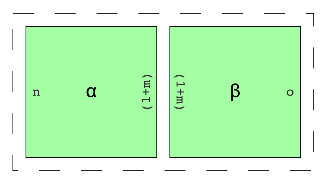
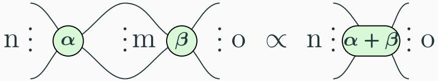


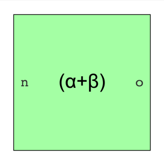
Diagram now corresponds to the IH



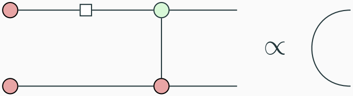
2. Inductive proof: Absolute Fusion



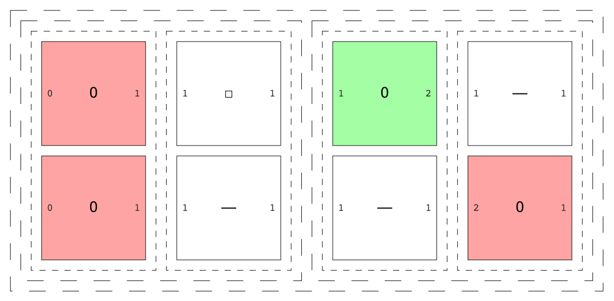
Qed.



3. Diagrammatic proof: Bell pair preparation

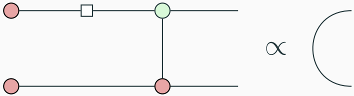


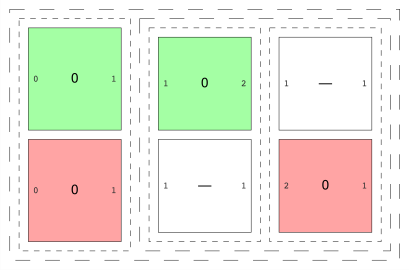
The lemma:



Let's first swap colors on the left

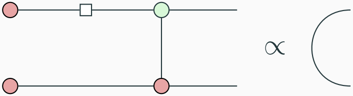
3. Diagrammatic proof: Bell pair preparation

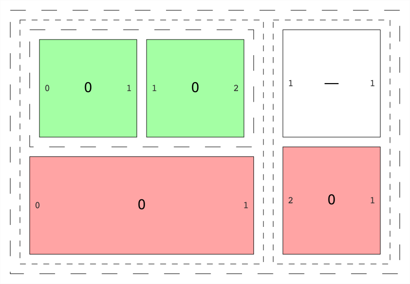




Let's reassociate!

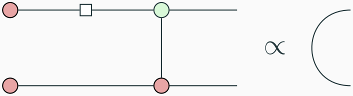
3. Diagrammatic proof: Bell pair preparation

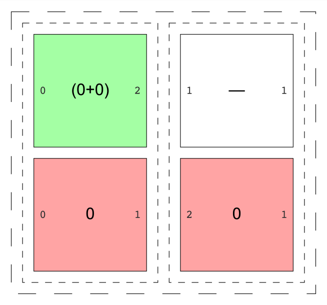




Let's fuse

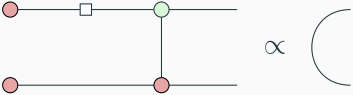
3. Diagrammatic proof: Bell pair preparation

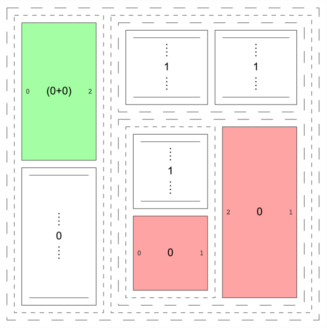




Separate the red node to fuse

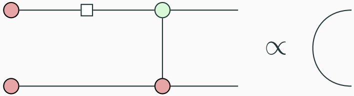
3. Diagrammatic proof: Bell pair preparation

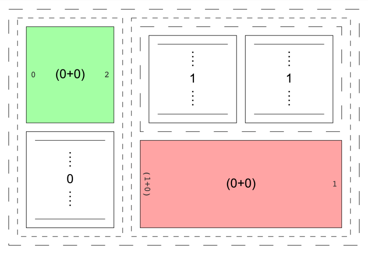




Let's fuse the red spiders

3. Diagrammatic proof: Bell pair preparation





Apply identity rule, remove wires

3. Diagrammatic proof: Bell pair preparation

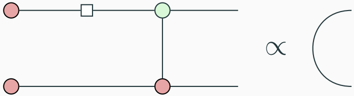
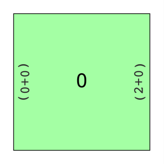
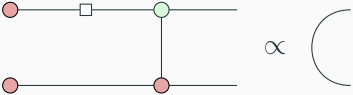


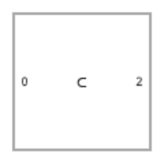
Diagram now corresponds to the a cap



3. Diagrammatic proof: Bell pair preparation



Qed.



Quantum circuit ingestion from RzQ gate set

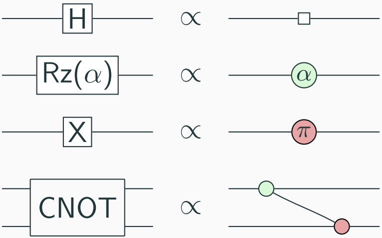
$$\frac{\mathsf{n} < \mathsf{q} \,:\, \mathbb{N}}{\texttt{H}\ \mathsf{n} \,:\, \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} \,:\, \mathbb{N}}{\texttt{X}\ n \,:\, \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} \,:\, \mathbb{N} \qquad \alpha \,:\, \mathbb{R}}{\texttt{Rz}(\alpha)\ n \,:\, \mathsf{Circuit\ q\ q}}$$

$$\frac{\mathsf{c,n} < \mathsf{q} \,:\, \mathbb{N}}{\texttt{CNOT}\ c\ n \,:\, \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{q} \,:\, \mathbb{N} \qquad c_1, c_2 \,:\, \mathsf{Circuit\ q\ q}}{\texttt{Compose}\ c_1\ c_2 \,:\, \mathsf{Circuit\ q\ q}}$$

Quantum circuit ingestion from RzQ gate set

$$\frac{\mathsf{n} < \mathsf{q} : \mathbb{N}}{\mathtt{H}\ \mathsf{n} : \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} : \mathbb{N}}{\mathtt{X}\ n : \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} : \mathbb{N} \qquad \alpha : \mathbb{R}}{\mathtt{Rz}(\alpha)\ n : \mathsf{Circuit\ q\ q}}$$

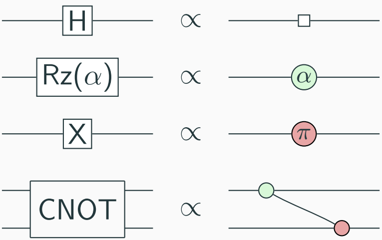
$$\frac{\mathsf{c,n} < \mathsf{q} : \mathbb{N}}{\mathtt{CNOT}\ c\ n : \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{q} : \mathbb{N} \qquad c_1, c_2 : \mathsf{Circuit\ q\ q}}{\mathtt{Compose}\ c_1\ c_2 : \mathsf{Circuit\ q\ q}}$$

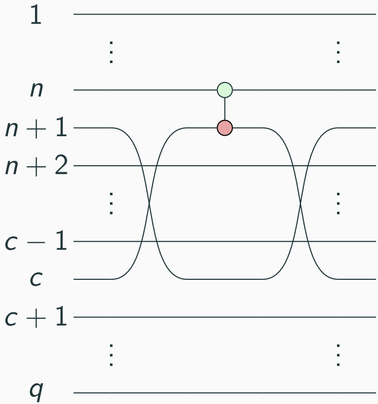


Quantum circuit ingestion from RzQ gate set

$$\frac{\mathsf{n} < \mathsf{q} \;:\; \mathbb{N}}{\mathtt{H}\ \mathsf{n} \;:\; \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} \;:\; \mathbb{N}}{\mathtt{X}\ n \;:\; \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{n} < \mathsf{q} \;:\; \mathbb{N} \qquad \alpha \;:\; \mathbb{R}}{\mathtt{Rz}(\alpha)\ n \;:\; \mathsf{Circuit\ q\ q}}$$

$$\frac{\mathsf{c{,}n} < \mathsf{q} \;:\; \mathbb{N}}{\mathtt{CNOT}\ c\ n \;:\; \mathsf{Circuit\ q\ q}} \qquad \frac{\mathsf{q} \;:\; \mathbb{N} \qquad c_1, c_2 \;:\; \mathsf{Circuit\ q\ q}}{\mathtt{Compose}\ c_1\ c_2 \;:\; \mathsf{Circuit\ q\ q}}$$





Discussion

- Any ZX diagram can be expressed
- Multiple ways to encode
- Deal with associativity information
- Dimensionality issues

How can I verify my graphical language?

- Find underlying categorical structure
- Formally extend structure
- Translate into proof assistant
- Deal with resulting associtativity issues

Future work

- Restore connection information
- Verify ZX-based compiler
- Prove ZX results

Summary

- Defined ZX diagrams inductively
- Inspired by string diagrams
- Multiple proof strategies
- ZX calculus is interesting!

Find *Vy*ZX on GitHub

https://github.com/inQWIRE/VyZX

arXiv

Coming soon...

References

- Bob Coecke and Aleks Kissinger, *Picturing quantum processes: A first course in quantum theory and diagrammatic reasoning*, Cambridge University Press, 2017.
- Jonathan Castello, Patrick Redmond, and Lindsey Kuper, *Inductive diagrams for causal reasoning*, 2023.
- Kesha Hietala, Robert Rand, Shih-Han Hung, Xiaodi Wu, and Michael Hicks, *A verified optimizer for quantum circuits*, Proc. ACM Program. Lang. **5** (2021), no. POPL.
- John van de Wetering, *Zx-calculus for the working quantum computer scientist*, 2020.