ZQ

Hector Miller-Bakewell

University of Oxford

2019-11-11

Things to look at:

- What is ZQ?
- Why is it interesting?
- Why does it look like this?

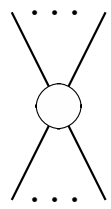
What is ZQ?

Generators

ZQ diagrams are open, directed, multi graphs made from

Generators

ZQ diagrams are open, directed, multi graphs made from

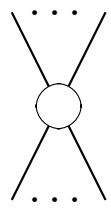


Generators

ZQ diagrams are open, directed, multi graphs made from

Generators

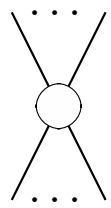
ZQ diagrams are open, directed, multi graphs made from



λ_c

Generators

ZQ diagrams are open, directed, multi graphs made from



λ_c

Generators

ZQ diagrams are open, directed, multi graphs made from

λ_c

(Really it's a PROP, but we're doing Only Topology Matters)

Quaternions (Quick Recap)

Quaternions are a non-commutative real algebra:

Quaternions (Quick Recap)

Quaternions are a non-commutative real algebra:

$$\mathbb{R} + \mathbb{R}i + \mathbb{R}j + \mathbb{R}k$$

Quaternions (Quick Recap)

Quaternions are a non-commutative real algebra:

$$\mathbb{R} + \mathbb{R}i + \mathbb{R}j + \mathbb{R}k$$

$$i^2 = j^2 = k^2 = ijk = -1$$

Unit quaternions and SU2

If we write a unit quaternion as

$$q_w + iq_x + jq_y + kq_z$$

Unit quaternions and SU2

If we write a unit quaternion as

$$q_w + iq_x + jq_y + kq_z$$

then there is a group isomorphism from unit quaternions to SU2:

$$\phi : q \mapsto \begin{pmatrix} q_w - iq_z & -q_y + iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix}$$

Unit quaternions and SO3

If we write a unit quaternion as an angle and a unit vector:

$$\cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}(iv_x + jv_y + kv_z)$$

Unit quaternions and SO3

If we write a unit quaternion as an angle and a unit vector:

$$\cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}(iv_x + jv_y + kv_z)$$

then there is a 2:1 group homomorphism to $SO3$:

$$\psi : q \mapsto \text{rotation by angle } \alpha \text{ about } v$$

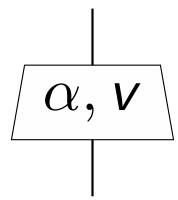
(q and $-q$ are sent to the same rotation)

Unit quaternions in two different forms

$$(\alpha, v) := \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}(iv_x + jv_y + kv_z)$$

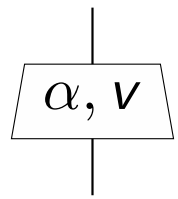
Unit quaternions in two different forms

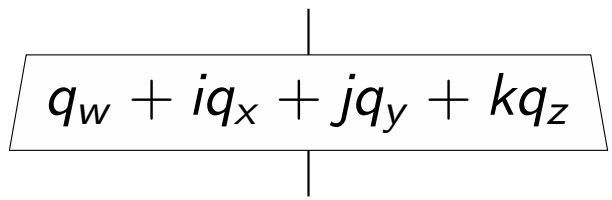
$$(\alpha, v) := \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}(iv_x + jv_y + kv_z)$$



Unit quaternions in two different forms

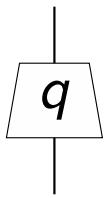
$$(\alpha, v) := \cos\frac{\alpha}{2} + \sin\frac{\alpha}{2}(iv_x + jv_y + kv_z)$$





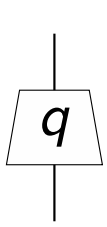
Interpreting q

$$\left[\!\!\left[\; \right.\right. \; \left.\left. \right]\!\!\right] = \begin{pmatrix} q_w - iq_z & -q_y + iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix}$$

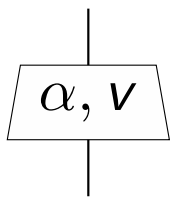


Interpreting q

$$\left[\!\!\left[\; \right.\right. \; \left.\left. \right]\!\!\right] = \begin{pmatrix} q_w - iq_z & -q_y + iq_x \\ -q_y - iq_x & q_w + iq_z \end{pmatrix}$$



$$\left[\!\!\left[\; \right.\right. \; \left.\left. \right]\!\!\right] = \begin{pmatrix} \cos\frac{\alpha}{2} - i\sin\frac{\alpha}{2}v_z & -i\sin\frac{\alpha}{2}(v_x + iv_y) \\ -i\sin\frac{\alpha}{2}(v_x - iv_y) & \cos\frac{\alpha}{2} + i\sin\frac{\alpha}{2}v_z \end{pmatrix}$$



Familiar faces

Pauli matrices!

Familiar faces

Pauli matrices!

$$\left[\!\left[\ \boxed{\pi, x}\ \right]\!\right] = -i \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\left[\!\left[\ \boxed{\pi, y}\ \right]\!\right] = -i \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}$$

$$\left[\!\left[\ \boxed{\pi, z}\ \right]\!\right] = -i \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Familiar faces

Hadamard

$$\left[\!\!\left[\quad \boxed{\pi, \tfrac{1}{\sqrt{2}}(x+z)} \quad \right]\!\!\right] = \frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Familiar faces

Hadamard

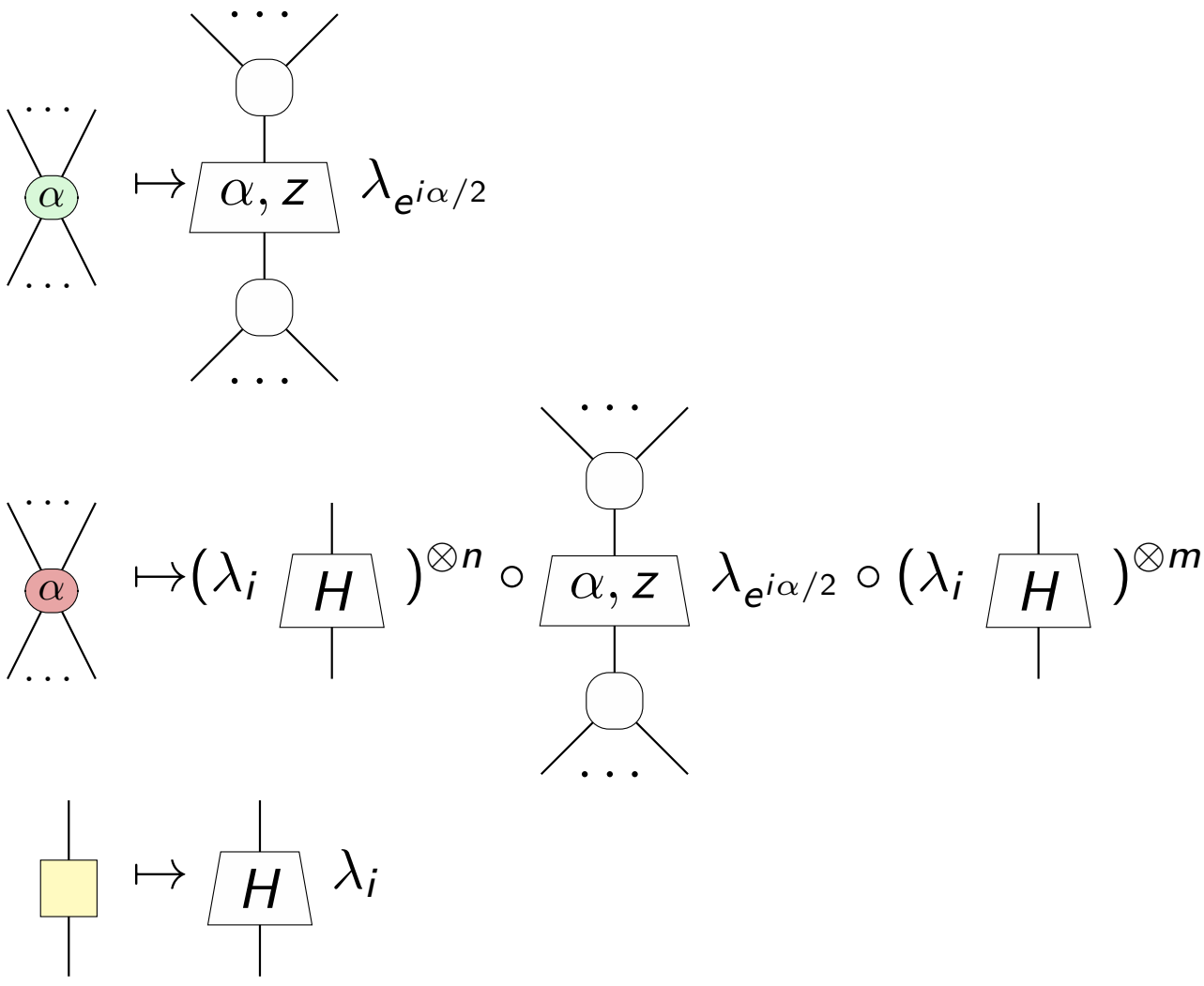
$$\left[\!\!\left[\; \pi, \tfrac{1}{\sqrt{2}}(x+z) \;\right]\!\!\right] = \frac{-i}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$:= \quad \pi, \tfrac{1}{\sqrt{2}}(x+z)$$

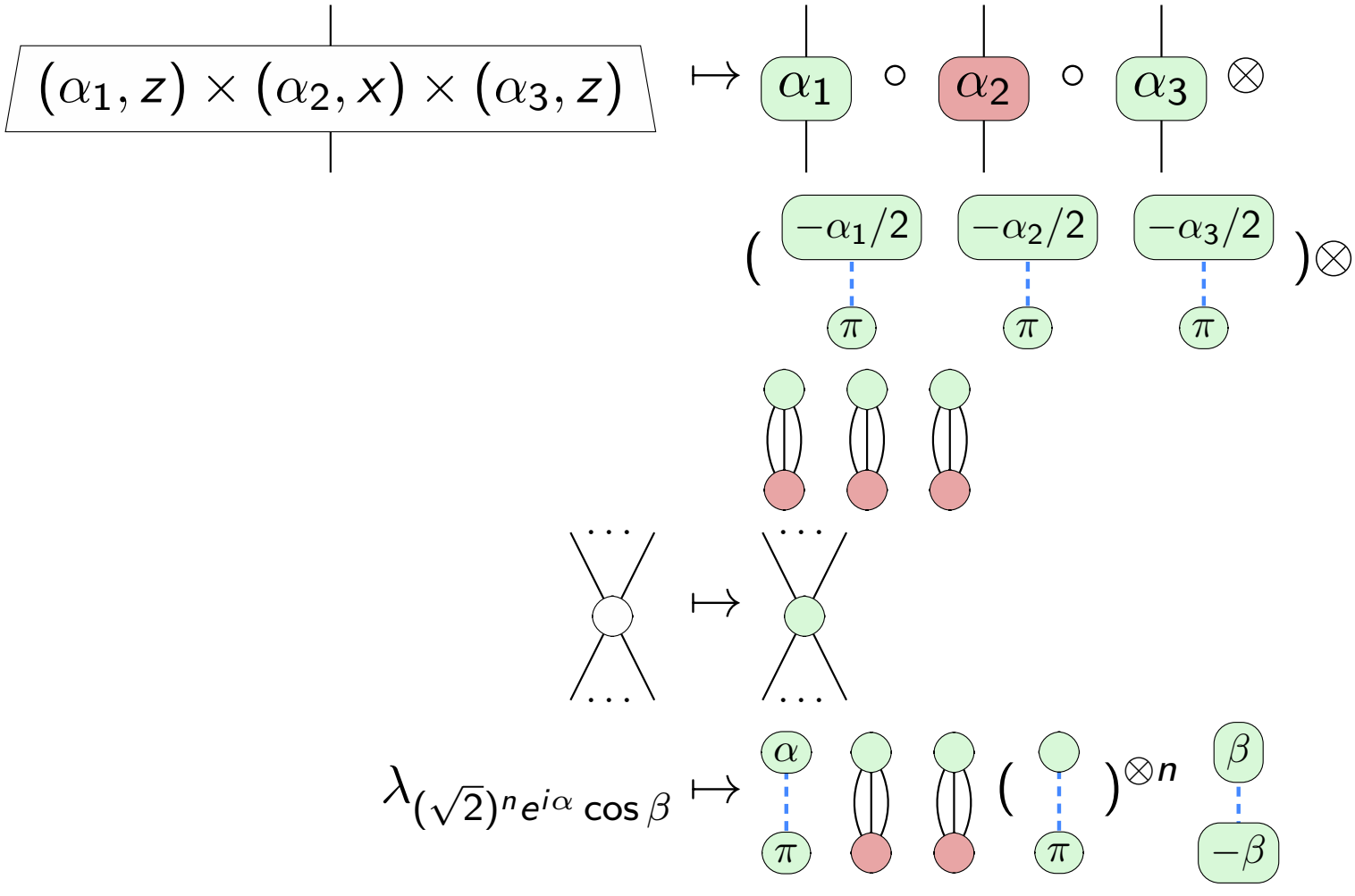
Completeness

ZQ is complete

Completeness



Completeness

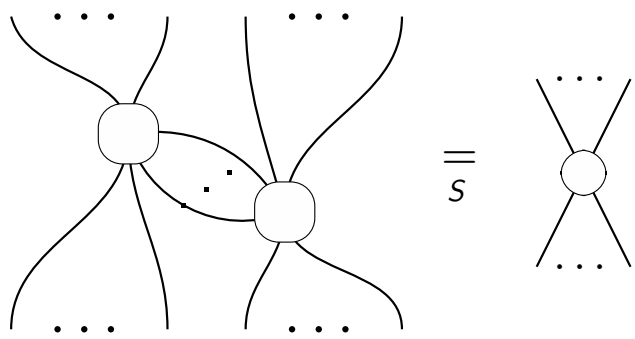


Rules of ZQ

So what are the rules?

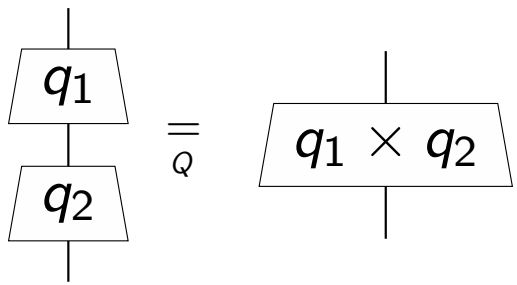
Rules of ZQ

Z spiders merge



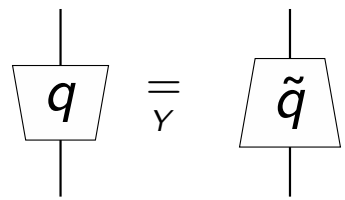
Rules of ZQ

Quaternions multiply



Rules of ZQ

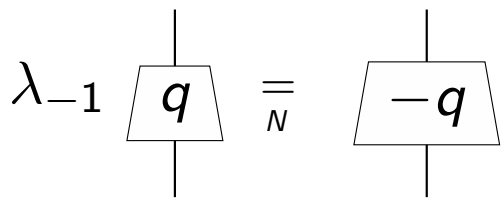
Y reflection / edge reversal



$$q_w + iq_x + jq_y + kq_z \mapsto q_w + iq_x - jq_y + kq_z$$

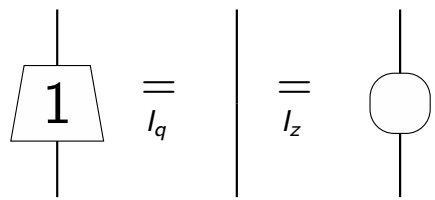
Rules of ZQ

Negation



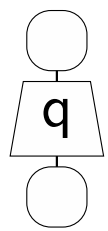
Rules of ZQ

Wire identities



Rules of ZQ

Action on a plus state



$\underset{A}{=} \lambda_{2(q_w - iq_x)}$

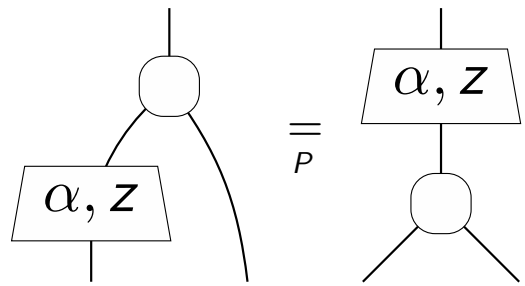
Rules of ZQ

Scalars are scalars

$$\lambda_x \lambda_y \underset{M}{=} \lambda_{x \times y} \qquad\qquad \lambda_1 \underset{I_1}{=}$$

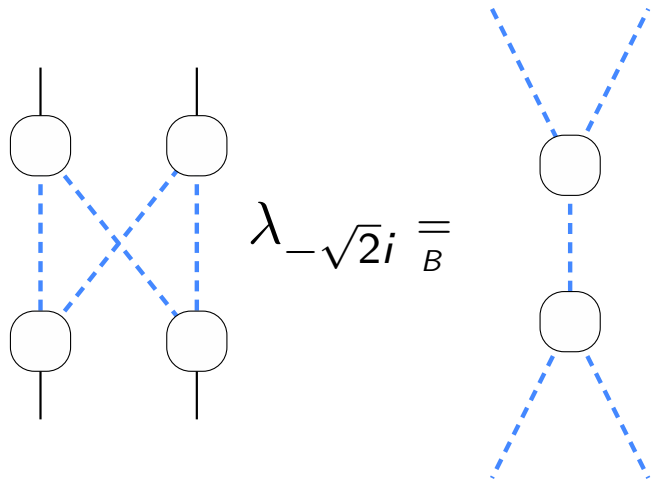
Rules of ZQ

Z phases



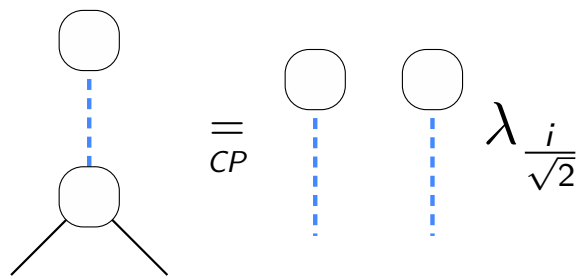
Rules of ZQ

Bialgebra



Rules of ZQ

Copy



Let's do that again, without scalars

Let's do that again, without scalars

And identifying $q \sim -q$

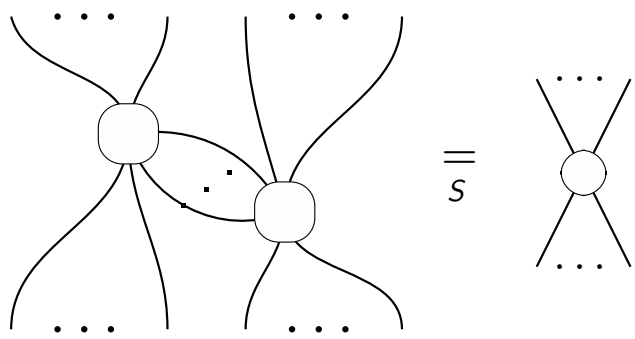
Let's do that again, without scalars

And identifying $q \sim -q$

So actually now using $SO3$ not $SU2$

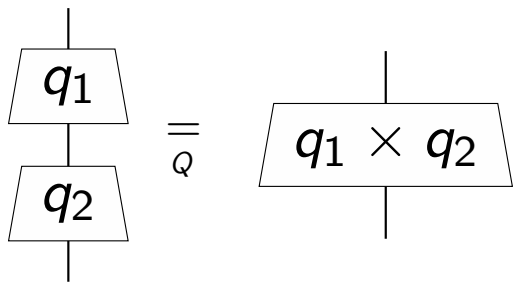
Scalar-free Rules of ZQ

Z spiders merge



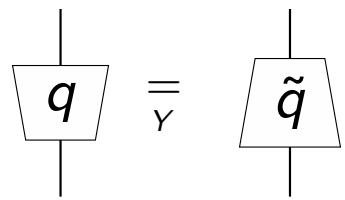
Scalar-free Rules of ZQ

Quaternions multiply



Scalar-free Rules of ZQ

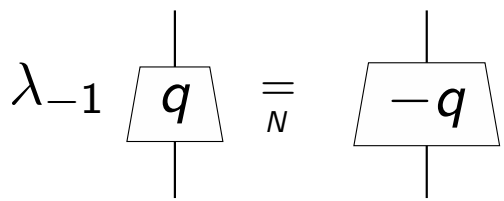
Y reflection / edge reversal



$$q_w + iq_x + jq_y + kq_z \mapsto q_w + iq_x - jq_y + kq_z$$

Scalar-free Rules of ZQ

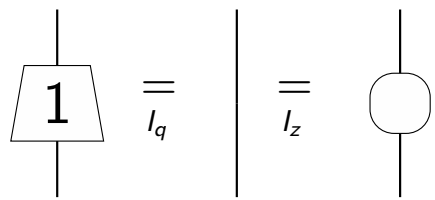
Negation



No longer needed.

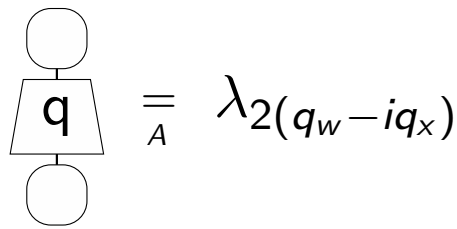
Scalar-free Rules of ZQ

Wire identities



Scalar-free Rules of ZQ

Action on a plus state



$\underset{A}{=} \quad \lambda_{2(q_w - iq_x)}$

No longer needed

Scalar-free Rules of ZQ

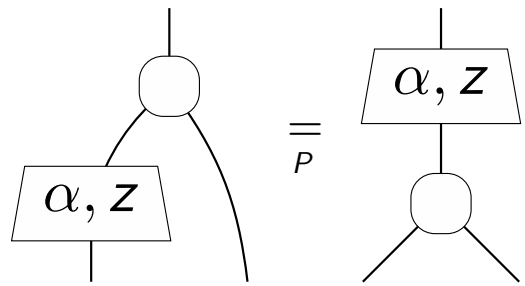
Scalars are scalars

$$\lambda_x \lambda_y \underset{M}{=} \lambda_{x \times y} \qquad\qquad \lambda_1 \underset{I_1}{=}$$

No longer needed

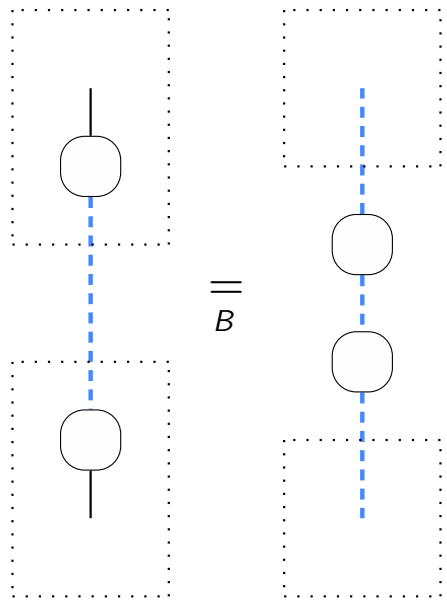
Scalar-free Rules of ZQ

Z phases

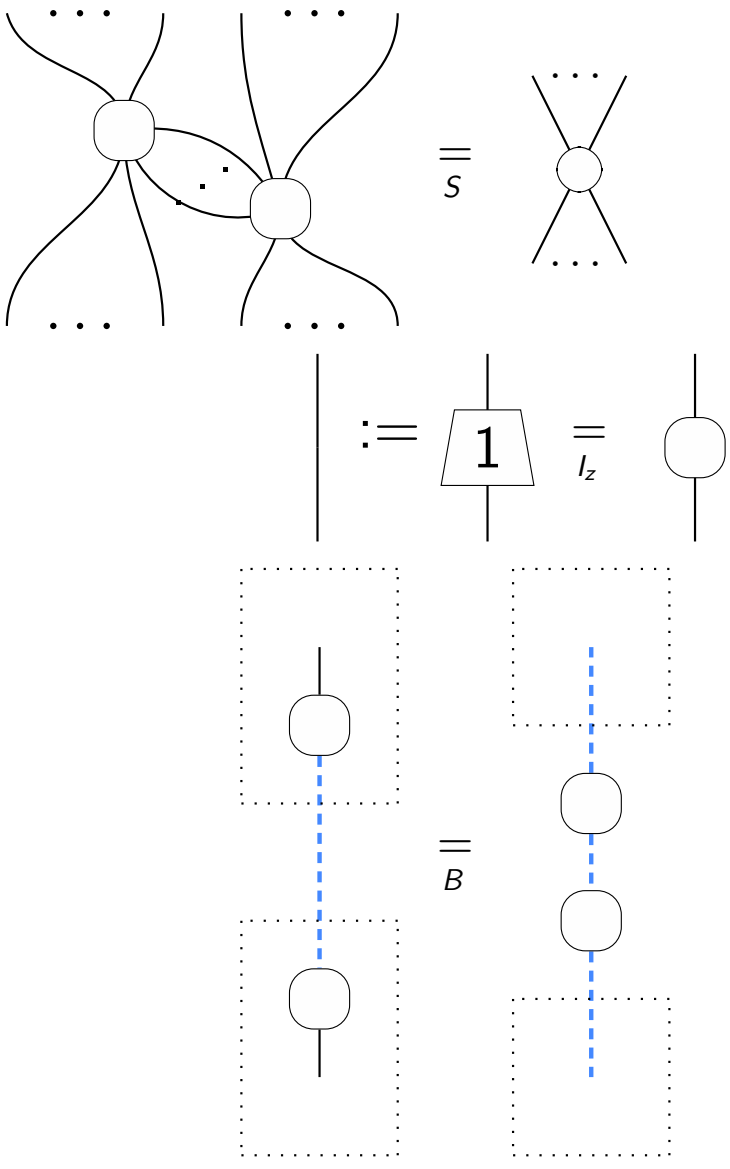


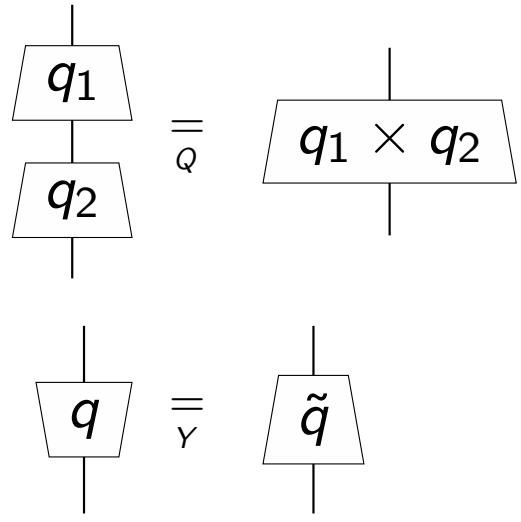
Scalar-free Rules of ZQ

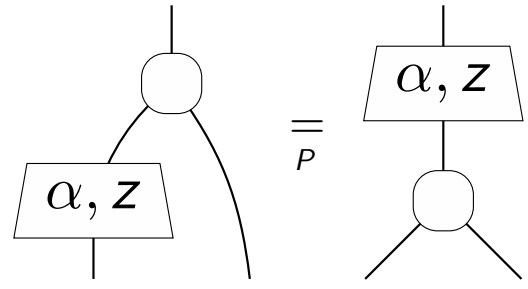
Bialgebra



Scalar-free Rules of ZQ







Why is it interesting?

Why is it interesting?

- Physics: Actually uses all rotations of the Bloch Sphere

Why is it interesting?

- Physics: Actually uses all rotations of the Bloch Sphere
- Structural: Uses directed edges *and* vertices, not just vertices

Why is it interesting?

- Physics: Actually uses all rotations of the Bloch Sphere
- Structural: Uses directed edges *and* vertices, not just vertices
- Phase algebra: Uses a group larger than $[0, 2\pi)$ (and a non-commutative one at that)

Why is it interesting?

- Physics: Actually uses all rotations of the Bloch Sphere
- Structural: Uses directed edges *and* vertices, not just vertices
- Phase algebra: Uses a group larger than $[0, 2\pi)$ (and a non-commutative one at that)
- Implementation: TriQ accredits part of its optimisation speedup to its use of quaternions

TriQ

Full-stack, real-system quantum computer studies: architectural comparisons and design insights (ISCA 2019)

Our work performs a full-stack, benchmark-driven hardware- software analysis of QC systems... We use our toolflow, TriQ, to conduct real-system measurements on seven running QC prototypes from three different groups, IBM, Rigetti, and University of Maryland. Overall, we demonstrate that leveraging microarchitecture details in the compiler improves program success rate up to 28x on IBM (geomean 3x), 2.3x on Rigetti (geomean 1.45x), and 1.47x on UMDTI (geomean 1.17x), compared to vendor toolflows.

TriQ

Full-stack, real-system quantum computer studies: architectural comparisons and design insights (ISCA 2019)

```
TriQ composes rotation operations by multiplying the
corresponding quaternions and creates a single
arbitrary rotation ...  Furthermore, on all three
vendors, Z-axis rotations are special operations that
are implemented in classical hardware and are
therefore error-free.  TriQ expresses the multiplied
quaternion as a series of two Z-axis rotations and
one rotation along either X or Y axis, thereby
maximizing the number of error-free operations.
```

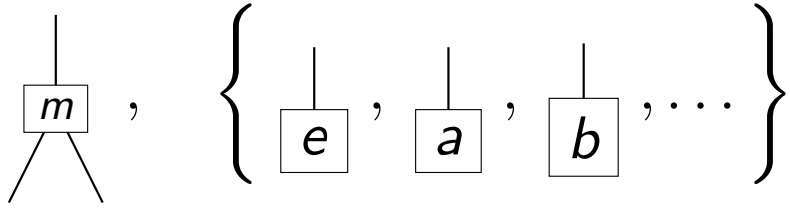
Why does it look like this?

Phase algebra

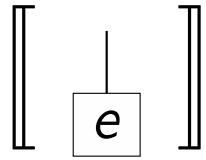
Consider a monoid over states in Qubit

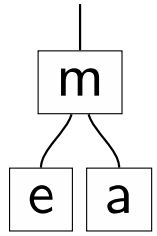
Phase algebra

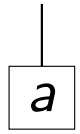
Consider a monoid over states in Qubit



Monoid spanning 0 dimensions

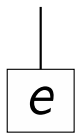
If 

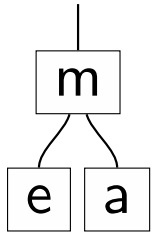
$= 0$ then because 

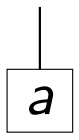
$=$ 

,

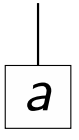
Monoid spanning 0 dimensions

If $[\![$ 

$]\!] = 0$ then because 

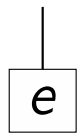
$=$ 

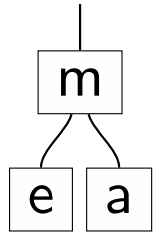
,

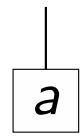
$[\![$ 

$]\!] = 0 \quad \forall a$

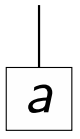
Monoid spanning 0 dimensions

If $[\![$ 

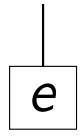
$]\!] = 0$ then because 

$=$ 

,

$[\![$ 

$]\!] = 0 \quad \forall a$

So from here on assume $[\![$ 

$]\!] = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Monoid spanning 1 dimension

If *span* $\left\{ \left[\!\!\left[\; \begin{array}{c} \\ \vert \\ \boxed{e} \end{array} \; \right]\!\!\right], \left[\!\!\left[\; \begin{array}{c} \\ \vert \\ \boxed{a} \end{array} \; \right]\!\!\right], \left[\!\!\left[\; \begin{array}{c} \\ \vert \\ \boxed{b} \end{array} \; \right]\!\!\right], \ldots \right\} = \left\{ \begin{pmatrix} \lambda \\ 0 \end{pmatrix} \right\}_{\lambda \in \mathbb{C}}$, then

Monoid spanning 1 dimension

If *span* $\left\{ [\![\; \boxed{e} \;]\!], [\![\; \boxed{a} \;]\!], [\![\; \boxed{b} \;]\!], \ldots \right\} = \left\{ \begin{pmatrix} \lambda \\ 0 \end{pmatrix} \right\}_{\lambda \in \mathbb{C}}$, then

$$[\![\; \boxed{m} \;]\!] = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot \end{pmatrix}$$

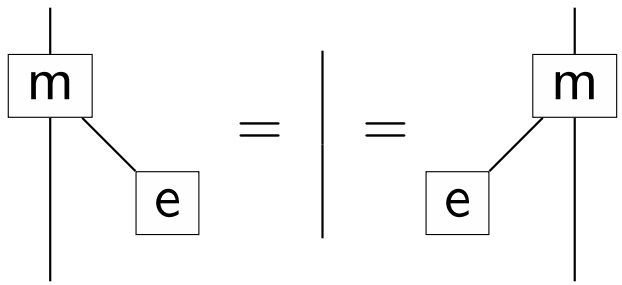
Monoid spanning 1 dimension

If *span* $\left\{ \left[\!\!\left[\begin{array}{c} \\ \boxed{e} \end{array} \right]\!\!\right], \left[\!\!\left[\begin{array}{c} \\ \boxed{a} \end{array} \right]\!\!\right], \left[\!\!\left[\begin{array}{c} \\ \boxed{b} \end{array} \right]\!\!\right], \ldots \right\} = \left\{ \begin{pmatrix} \lambda \\ 0 \end{pmatrix} \right\}_{\lambda \in \mathbb{C}}$, then

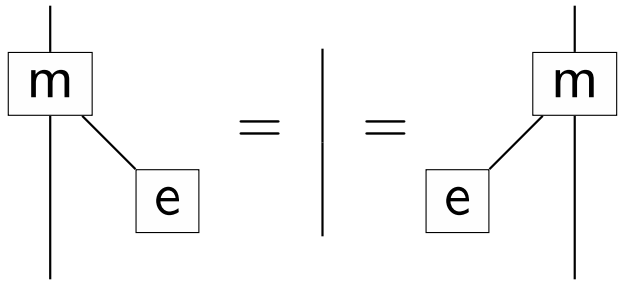
$$\left[\!\!\left[\begin{array}{c} \\ \boxed{m} \end{array} \right]\!\!\right] = \begin{pmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot \end{pmatrix}$$

$$\left[\!\!\left[\begin{array}{c} \\ \boxed{\text{m}} \\ \boxed{\text{a}}\ \boxed{\text{b}} \end{array} \right]\!\!\right] = \begin{pmatrix} \lambda_a \lambda_b \\ 0 \end{pmatrix} = \left[\!\!\left[\begin{array}{c} \\ \boxed{\text{m}} \\ \boxed{\text{b}}\ \boxed{\text{a}} \end{array} \right]\!\!\right]$$

Monoid spanning all of $\mathbb{C}^2$



Monoid spanning all of $\mathbb{C}^2$



Forcing the interpretation

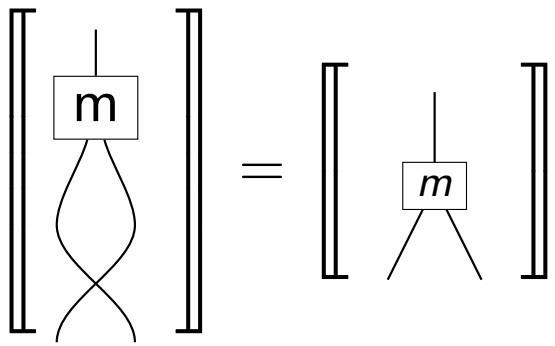
$$\left[\!\!\left[\quad \begin{array}{c} \\ m \\ \diagup\ \diagdown \end{array} \quad \right]\!\!\right] = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 1 & y \end{pmatrix}$$

Unital implies commutative

So in all qubit models of monoids acting on states, that monoid is commutative

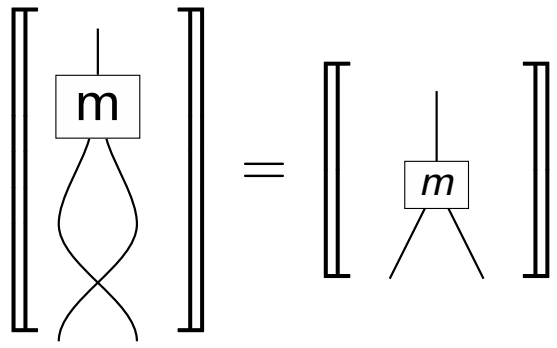
Unital implies commutative

So in all qubit models of monoids acting on states, that monoid is commutative



Unital implies commutative

So in all qubit models of monoids acting on states, that monoid is commutative



So we could never represent SO3 this way

Next time

Aside: This line of reasoning has led to a classification of phase rings acting on states over K-bits, but more on that soon

Conclusion

Conclusion

- ZQ exists and is complete

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere
- It reflects optimisation strategies used in the wild

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere
- It reflects optimisation strategies used in the wild
- It explores phase algebras beyond $[0, 2\pi)$

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere
- It reflects optimisation strategies used in the wild
- It explores phase algebras beyond $[0, 2\pi)$
- It has a different structure to ZW / ZX / ZH

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere
- It reflects optimisation strategies used in the wild
- It explores phase algebras beyond $[0, 2\pi)$
- It has a different structure to ZW / ZX / ZH
- Probably not something to do in your head

Conclusion

- ZQ exists and is complete
- The scalar-free version is neat and tidy
- It reflects the full symmetry of the Bloch Sphere
- It reflects optimisation strategies used in the wild
- It explores phase algebras beyond $[0, 2\pi)$
- It has a different structure to ZW / ZX / ZH
- Probably not something to do in your head
- But compilers and proof assistants can implement it just fine

Thank you

Scalar-free Rules of ZQ