

Quantum software:
Phase-free ZX diagrams and CSS codes

Aleks Kissinger

February 16, 2024

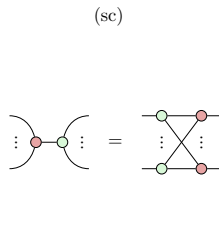
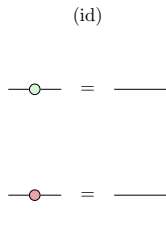
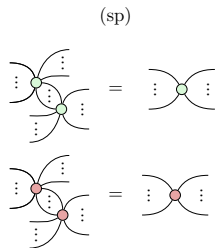
Phase-free ZX-diagrams

...are made of spiders with $\alpha = 0$:

$$\begin{array}{c} \vdots \\ \text{---} \circ \text{---} \\ \text{---} \end{array} \quad := \quad |0\dots 0\rangle\langle 0\dots 0| + |1\dots 1\rangle\langle 1\dots 1|$$

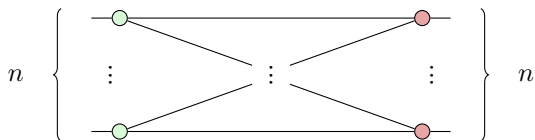
$$\begin{array}{c} \vdots \\ \text{---} \circ \text{---} \\ \text{---} \end{array} \quad := \quad |+\dots+\rangle\langle +\dots+| + |-\dots-\rangle\langle -\dots-|$$
$$= \quad N \sum_{\oplus; b_i=0} |b_1\dots b_n\rangle\langle b_{n+1}\dots b_{n+m}|$$

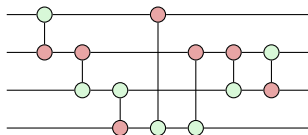
Phase-free ZX-calculus



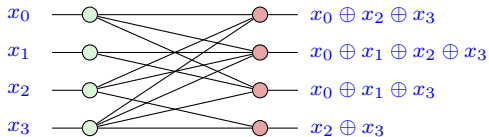
Unitaries

Unitary $\implies m = n, j = k = 0$



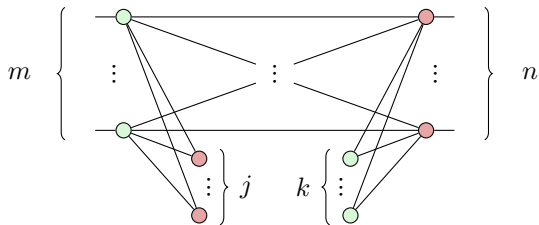


↓ *



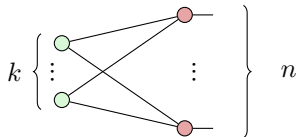
States

State $\implies m = 0, j = 0$



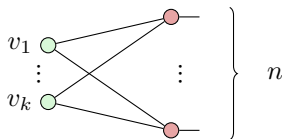
States

State $\implies m = 0, j = 0$



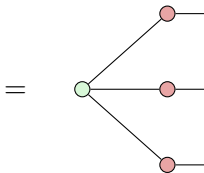
States

State $\implies m = 0, j = 0$



$$|\psi\rangle = \sum_{v \in S} |v\rangle \text{ where } S = \text{span}\{v_1, \dots, v_k\} \subseteq \mathbb{F}_2^n$$

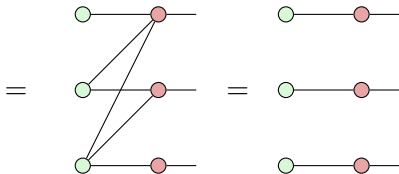
$$\begin{aligned}
 |\text{GHZ}\rangle &= |000\rangle + |111\rangle \\
 &= \sum_{v \in S} |v\rangle \quad \text{where } S = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}
 \end{aligned}$$



$$\begin{aligned}
 |+++ \rangle &= \sum_{v \in \mathbb{F}_2^3} |v\rangle \\
 &= \sum_{v \in S} |v\rangle \quad \text{where } S = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}
 \end{aligned}$$

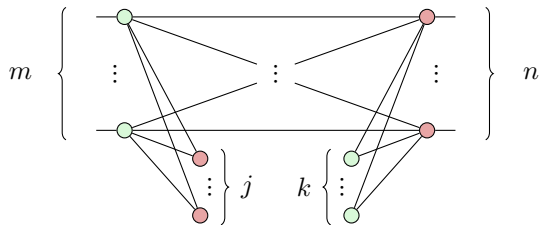


$$\begin{aligned}
 |+++ \rangle &= \sum_{v \in \mathbb{F}_2^3} |v\rangle \\
 &= \sum_{v \in S} |v\rangle \quad \text{where } S = \text{span} \left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}
 \end{aligned}$$



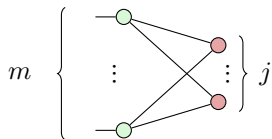
Effects

Effect $\implies n = 0, k = 0$



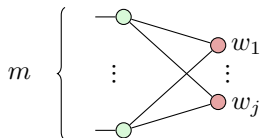
Effects

Effect $\implies n = 0, k = 0$



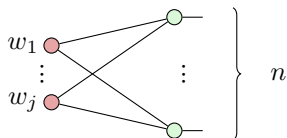
Effects

Effect $\implies n = 0, k = 0$



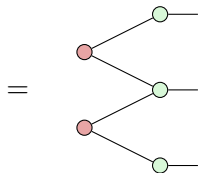
$$\langle \phi | = \sum_{v \in S} \langle v | \quad \text{where } S^\perp = \text{span}\{w_1, \dots, w_j\} \subseteq \mathbb{F}_2^m$$

Or a second way to write states...



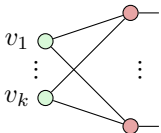
$$|\psi\rangle = \sum_{v \in S} |v\rangle \quad \text{where } S^\perp = \text{span}\{w_1, \dots, w_j\} \subseteq \mathbb{F}_2^n$$

$$\begin{aligned}
|\text{GHZ}\rangle &= |000\rangle + |111\rangle \\
&= \sum_{v \in S} |v\rangle \quad \text{where } S = \{v \mid v_1 \oplus v_2 = 0, v_2 \oplus v_3 = 0\} \\
&= \sum_{v \in S} |v\rangle \quad \text{where } S^\perp = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}
\end{aligned}$$



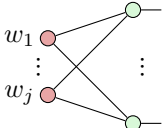
Theorem

A state represented by a phase-free ZX-diagram is uniquely fixed by a subspace $S \subseteq \mathbb{F}_2^n$ (or equivalently $S^\perp \subseteq F_2^n$).



A ZX-diagram with two columns of nodes. The left column has green nodes labeled v_1, \dots, v_k . The right column has red nodes. Each green node is connected to both red nodes. Vertical ellipses are present between the green nodes and between the red nodes.

$$= \sum_{v \in S} |v\rangle \text{ where } S = \text{span}\{v_1, \dots, v_k\}$$



A ZX-diagram with two columns of nodes. The left column has red nodes labeled w_1, \dots, w_j . The right column has green nodes. Each red node is connected to both green nodes. Vertical ellipses are present between the red nodes and between the green nodes.

$$= \sum_{v \in S} |v\rangle \text{ where } S^\perp = \text{span}\{w_1, \dots, w_j\}$$

Stabiliser Theory

Theorem (FTST)

If \mathcal{S} has k generators, then $\text{Stab}(\mathcal{S})$ is a 2^{n-k} dimensional subspace of $(\mathbb{C}^2)^{\otimes n}$.

$$\begin{array}{ccc} k = n & \implies & \text{Stab}(\mathcal{S}) = \{\lambda|\psi\rangle \mid \lambda \in \mathbb{C}\} \\ \textit{maximal} & & \textit{1D subspace} \end{array}$$

CSS codes

Definition

For $S \subseteq \mathbb{F}_2^n$, $T \subseteq S^\perp$, a **CSS code** is a stabiliser group with generators:

$$\vec{X}_i := \bigotimes_{q=1}^{\dim S} X^{(v_i)_q} \qquad \vec{Z}_j := \bigotimes_{q=1}^{\dim T} Z^{(w_j)_q}$$

where $S = \text{span}\{v_i\}$ and $T = \text{span}\{w_i\}$.

A CSS code is maximal iff $T = S^\perp$, i.e. it has generators:

$$\vec{X}_j := X^{(v_j)_1} \otimes \dots \otimes X^{(v_j)_n} \qquad \vec{Z}_j := Z^{(w_j)_1} \otimes \dots \otimes Z^{(w_j)_n}$$

where $S = \text{span}\{v_j\}$ and $S^\perp = \text{span}\{w_j\}$.

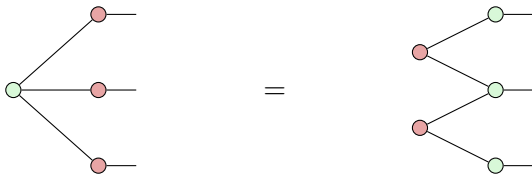
Example

The stabiliser group of $|\text{GHZ}\rangle$ is generated by:

$$X \otimes X \otimes X \quad Z \otimes Z \otimes I \quad I \otimes Z \otimes Z$$

This is a maximal CSS code, where:

$$S = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\} \quad S^\perp = \text{span} \left\{ \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\}$$

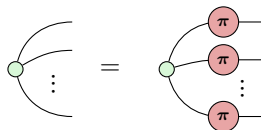


Theorem

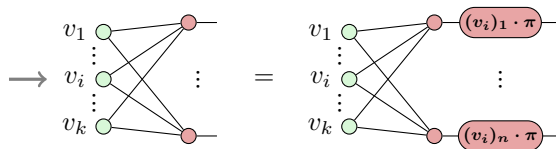
The ZX-diagram associated with $S \subseteq \mathbb{F}_2^n$ is the unique stabiliser state of the maximal CSS code defined by (S, S^\perp) .

Proof

Using:



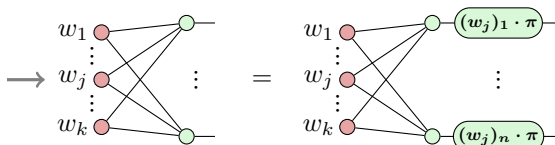
compute the X-stabilisers by “firing” each basis vector of S :



$$|\psi\rangle = (X^{(v_i)_1} \otimes \dots \otimes X^{(v_i)_n})|\psi\rangle$$

Proof (cont'd)

Similarly, compute the Z-stabilisers from S^\perp :



$$|\psi\rangle = (Z^{(w_j)_1} \otimes \dots \otimes Z^{(w_j)_n})|\psi\rangle$$

This gives $\dim S + \dim S^\perp = n$ generators for n qubits, so $|\psi\rangle$ uniquely fixed by FTST. □

Corollary

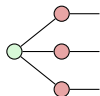
We can translate a maximal CSS code directly into a ZX-diagram in 2 ways.

For example, $\{X \otimes X \otimes X, Z \otimes Z \otimes I, I \otimes Z \otimes Z\}$ gives:

X-representation:

$$\{X \otimes X \otimes X\}$$

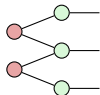
\rightsquigarrow



Z-representation:

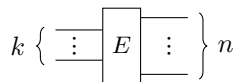
$$\{Z \otimes Z \otimes I, I \otimes Z \otimes Z\}$$

\rightsquigarrow



Quantum error correction

...is done by encoding some **logical** qubits into a bigger space of **physical** qubits:



E defines a **stabiliser code** when:

$$\text{Im} \left(\begin{array}{c} \text{---} \\ \vdots \\ \text{---} \end{array} \begin{array}{|c|} \hline E \\ \hline \end{array} \begin{array}{c} \text{---} \\ \vdots \\ \text{---} \end{array} \right) = \text{Stab}(\mathcal{S})$$

where \mathcal{S} is a stabiliser group with $n - k$ generators.

Quantum error correction

We can detect errors without destroying the state by measuring stabilisers in \mathcal{S} .

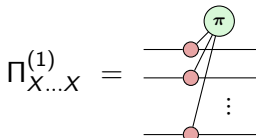
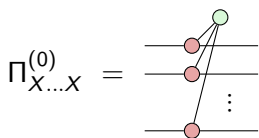
For CSS codes, 2 kinds of stabiliser measurements are relevant:

$$\mathcal{M}_{X\dots X} := \{\Pi_{X\dots X}^{(0)}, \Pi_{X\dots X}^{(1)}\}$$

$$\mathcal{M}_{Z\dots Z} := \{\Pi_{Z\dots Z}^{(0)}, \Pi_{Z\dots Z}^{(1)}\}$$

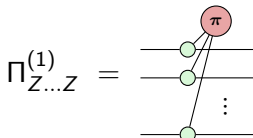
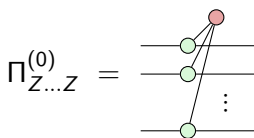
X measurements

$$\mathcal{M}_{X\dots X} = \left\{ \Pi_{X\dots X}^{(k)} := \frac{1}{2} (I + (-1)^k X \otimes \dots \otimes X) \right\}$$



Z measurements

$$\mathcal{M}_{Z\dots Z} = \left\{ \Pi_{Z\dots Z}^{(k)} := \frac{1}{2}(I + (-1)^k Z \otimes \dots \otimes Z) \right\}$$



Example

The GHZ code:

$$\mathcal{S} := \{\cancel{X \otimes X \otimes X}, Z \otimes Z \otimes I, I \otimes Z \otimes Z\}$$

Then:

$$\text{Im} \left(\text{---} \textcircled{\text{---}} \right) = \text{span}\{|000\rangle, |111\rangle\} = \text{Stab}(\mathcal{S})$$

Example

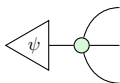
The GHZ code:

$$\mathcal{S} := \{\cancel{X \otimes X \otimes X}, Z \otimes Z \otimes I, I \otimes Z \otimes Z\}$$

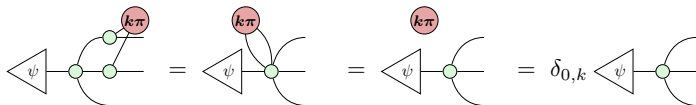
Then:

$$\text{Im} \left(\text{---} \textcircled{\text{green}} \text{---} \right) = \text{span}\{|000\rangle, |111\rangle\} = \text{Stab}(\mathcal{S})$$

So, we can encode states like this:



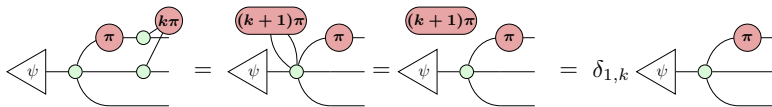
Applying Π_{ZZI}^\pm to an encoded state:



Hence:

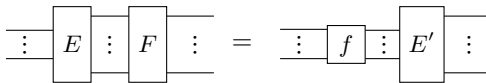
$$\text{Prob}_{ZZI}(k \mid \langle \psi | \text{---} \text{---} \rangle) = \delta_{0,k}$$

Applying Π_{ZZI}^\pm to an encoded state with an error:



Hence:

$$\text{Prob}_{ZZI} \left(k \mid \left\langle \psi \right| \begin{array}{c} \text{---} \text{---} \text{---} \\ \text{---} \text{---} \text{---} \\ \text{---} \text{---} \text{---} \end{array} \right) = \delta_{1,k}$$



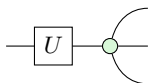
Logical operators

Note:

$$\text{Im} \left(\begin{array}{c} \text{---} \\ \vdots \\ \text{---} \end{array} \begin{array}{|c|} \hline E \\ \hline \end{array} \begin{array}{c} \text{---} \\ \vdots \\ \text{---} \end{array} \right) = \text{Stab}(\mathcal{S})$$

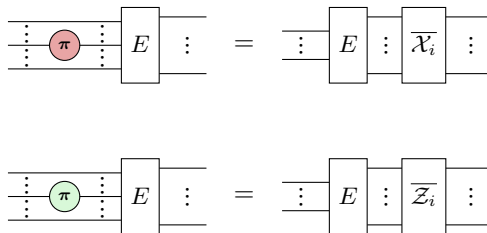
only fixes the **image** of E , not E itself.

For example, the following is also a GHZ encoder:



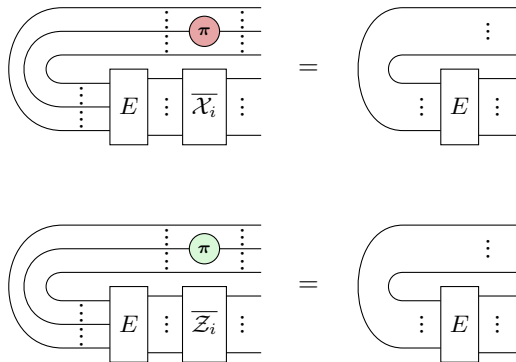
Logical operators

To fix E , we should fix $2k$ more **logical operators** by “pushing” Pauli X and Z ops through the encoder:



Logical operators

Equivalently, we fix $2k$ more stabilisers for the $n + k$ qubit state $|E\rangle := (I \otimes E)|U\rangle$:



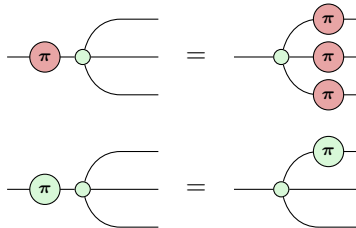
$(n - k) + 2k = n + k$ stabilisers for $|E\rangle$

Example

The GHZ code has stabilizers and logical operators:

$$\vec{Z}_1 = Z_1 Z_2 \quad \vec{Z}_2 = Z_2 Z_3$$

$$\vec{X} = X_1 X_2 X_3 \quad \vec{Z} = Z_1$$

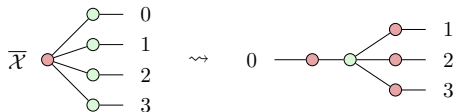


Stabilizers for $|E\rangle$:

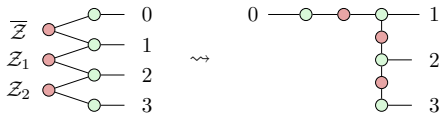
$$\vec{X}'_1 = X_1 X_2 X_3 \quad \vec{Z}'_1 = Z_1 Z_2 \quad \vec{Z}'_2 = Z_2 Z_3$$

$$\vec{X}' = X_0 X_1 X_2 X_3 \quad \vec{Z}' = Z_0 Z_1$$

X-representation:

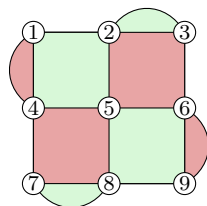


Z-representation:



The surface code

...is a 2D lattice of $d \times e$ qubits:



\rightsquigarrow

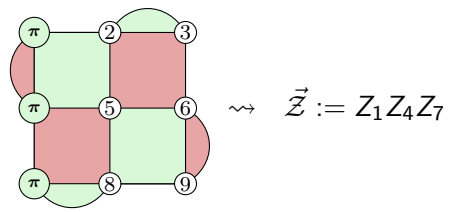
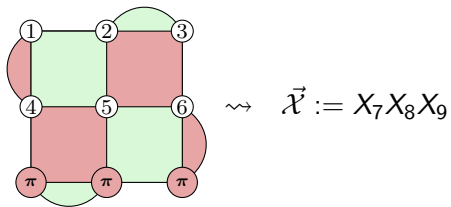
$$\vec{X}_1 := X_2 X_3 X_5 X_6 \quad \vec{X}_2 := X_4 X_5 X_7 X_8$$

$$\vec{X}_3 := X_1 X_4 \quad \vec{X}_4 := X_6 X_9$$

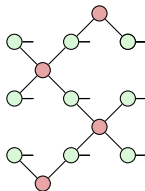
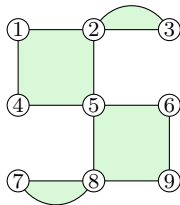
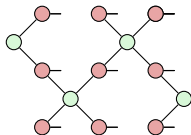
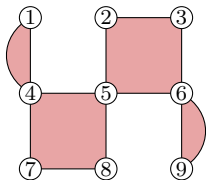
$$\vec{Z}_1 := Z_1 Z_2 Z_4 Z_5 \quad \vec{Z}_2 := Z_5 Z_6 Z_8 Z_9$$

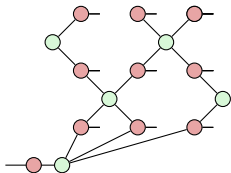
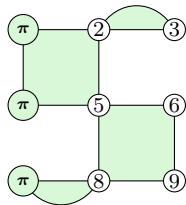
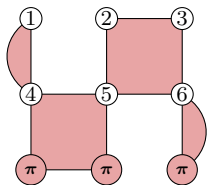
$$\vec{Z}_3 := Z_2 Z_3 \quad \vec{Z}_4 := Z_7 Z_8$$

$$(d-1)(e-1) + d - 1 + e - 1 = de - 1 \text{ stabilisers}$$

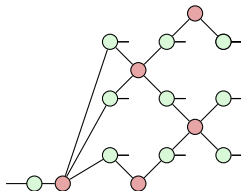


2 logical operators



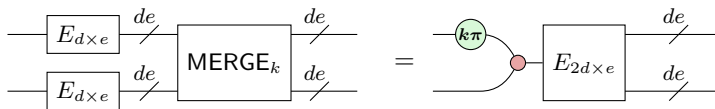
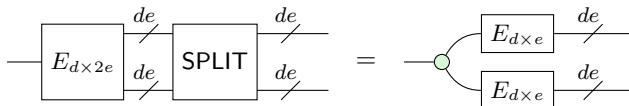


=

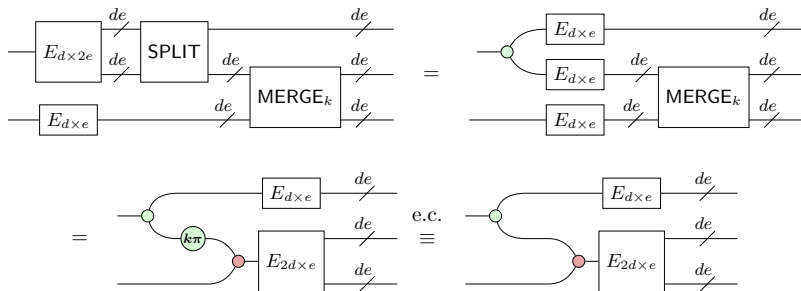


Lattice surgery

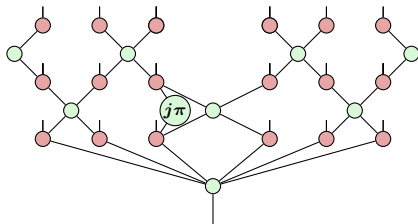
In the surface code, we can implement physical operations that behave like **SPLIT** and **MERGE** on logical qubits:



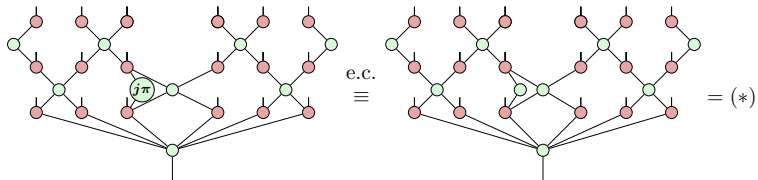
This lets us do entangling operations, e.g. CNOT:



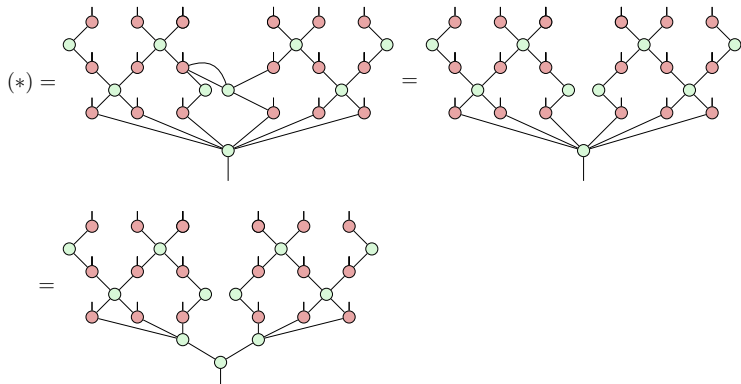
Split



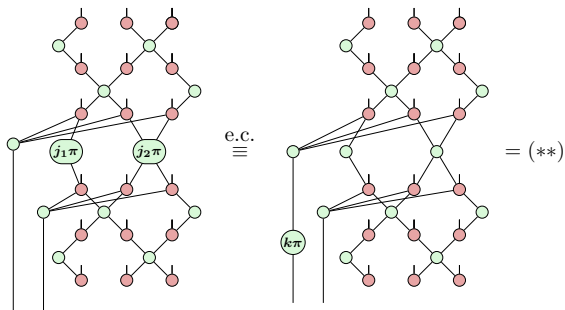
Split



Split

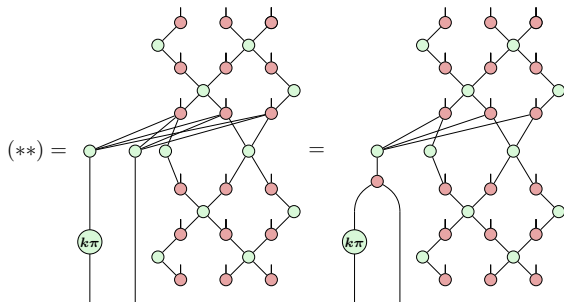


Merge



$$k := j_1 \oplus j_2$$

Merge



Final notes

This all used the X-representation. Flip to the Z-representation to get the colour-reversed split and merge.

Similar tricks implement:

- ▶ Entangled measurements
- ▶ Magic state injection
- ▶ \implies universal FTQC

Other CSS codes like colour codes translate to ZX very similarly. L.S. should pretty much work the same way.