

$\mathbb{Z}X$ -Calculus

\mathbb{Z} -Spiders

$$n \left\{ \begin{array}{c} \text{Diagram with } \alpha \\ \text{Diagram with } -\alpha \end{array} \right\}_m := \underbrace{|000\rangle\langle 00..0|}_m + e^{i\alpha} \underbrace{|111..\rangle\langle 11..1|}_n$$

\mathbb{X} -Spiders Recall $|+\rangle = \frac{1}{\sqrt{2}}(|1\rangle + |-\rangle)$ $|-\rangle = \frac{1}{\sqrt{2}}(|1\rangle - |-\rangle)$

$$\text{Diagram with } \alpha : = |++..+X++..+1 + e^{i\alpha} |--..-X--..-1$$

Thm: Any Linear map from \mathbb{C}^{2^n} to \mathbb{C}^{2^m}
can be written as a $\mathbb{Z}X$ -Diagram

But: Can also reason diagrammatically

ZX diagrams have "extreme" OCM.

They are invariant under:

- SWAPPING SPIDER-LEGS:

$$\cdot \quad \text{Diagram} = \text{Diagram} = \text{Diagram} = \dots$$

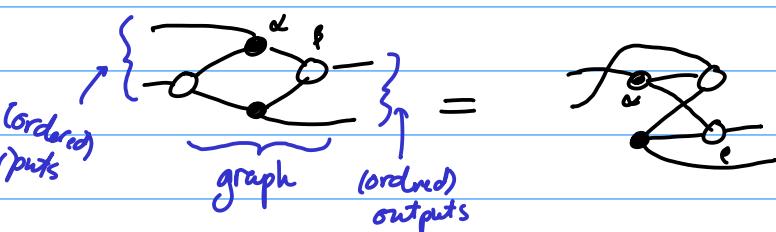
- CHANGING DIRECTION

$$\text{Diagram} = \text{Diagram}$$

$$(I \otimes X[\beta]_2^1)(Z[\alpha]_2^2 \otimes I) = (Z[\alpha]_3^1 \otimes I)(I \otimes X[\beta]_1^2)$$

\Rightarrow they can be treated as undirected graphs (ω lists of inputs & outputs)

e.g.



In addition, we have **rewrite rules**

We call this the **ZX -calculus**

(0) "WIRE" RULES

$$\text{---} = \text{---} \bullet = \text{---} \xrightarrow{\text{ID}}$$

$$(\text{---} := \alpha = \alpha) = (\text{---} := \beta = \beta)$$

$|00\rangle + |11\rangle$ $|++\rangle + |-+\rangle$

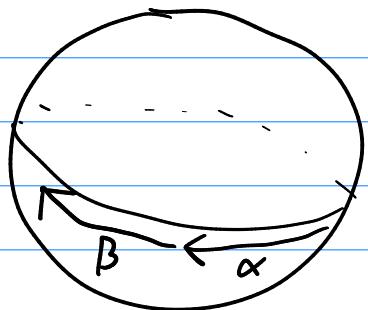
(1) SPIDER-FUSION

$$\text{---} \alpha \text{---} \bullet \text{---} \beta \text{---} = \text{---} \alpha + \beta \text{---}$$

$$\text{---} \alpha \text{---} \bullet \text{---} \beta \text{---} = \text{---} \alpha + \beta \text{---}$$

$$\text{---} \alpha \text{---} \bullet \text{---} \beta \text{---} = \text{---} \alpha + \beta \text{---}$$

$$\text{---} \alpha \text{---} \bullet \text{---} \alpha \text{---} = \text{---} \alpha \text{---} = \text{---}$$



Recall: $XOR \approx \bullet$

$$\begin{aligned} |0\rangle &\approx \text{---} \bullet \text{---} \\ |1\rangle &\approx \text{---} \pi \text{---} \end{aligned} \quad \left. \begin{aligned} \text{---} \pi \text{---} &\approx |a\rangle \quad a \in \{0, 1\} \end{aligned} \right\}$$

$$\text{---} \pi \text{---} = \text{---} \pi \text{---} = \text{---} \pi \text{---} \quad \text{---} \pi \text{---} = \text{---} 2\pi \text{---} = \text{---}$$

$$\text{---} a\pi \text{---} = \text{---} (a+b)\pi \text{---} = \text{---} (a \otimes b)\pi \text{---}$$

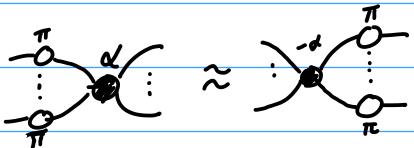
$$\text{COPY} = \text{---} \circ \text{---}$$

$|x\rangle \mapsto |xx\rangle$

$$\text{---} \circ \text{---} = \text{---} \circ \text{---} = \text{---} \circ \text{---}$$

associativity

(2) π -rule:



$$\overline{\overset{\pi}{\text{---}}} = |0\rangle\langle 0| - |1\rangle\langle 1| = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \text{ We call this the Pauli } Z$$

$$\overline{\overset{\pi}{\text{---}}} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ Pauli } X$$

$$\overline{\overset{\pi}{\text{---}}} = \overline{\overset{\pi}{\text{---}}} \quad \text{NOT gate}$$

$$\text{NOT}|0\rangle = |1\rangle$$

$$\overset{\pi}{\text{---}} = \overset{\pi}{\text{---}}$$

$$\text{Ex: } \overline{\overset{\pi}{\text{---}}} \overset{\alpha}{\text{---}} \overset{\pi}{\text{---}} \approx \overset{-\alpha}{\text{---}} \quad \overline{\overset{\pi}{\text{---}}} \overset{\pi}{\text{---}} \approx \overline{\overset{\pi}{\text{---}}} = \overline{\overset{\pi}{\text{---}}}$$

(3) Colour Change:

$$\overline{\overset{\square}{\text{---}}} = \overline{\overset{\square}{\text{---}}}$$

$$\text{where } \overline{\overset{\square}{\text{---}}} \approx \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

works because $H|0\rangle = |+\rangle$ $H|1\rangle = |- \rangle$ Hadamard

$$\text{Ex: } \overline{\overset{\square}{\text{---}}} \overset{\pi}{\text{---}} = \overline{\overset{\pi}{\text{---}}}$$

$$\overline{\overset{\square}{\text{---}}} = \overline{\text{---}}$$

$$H \otimes H = H \times H$$

$$\text{Note: } \overline{\overset{\square}{\text{---}}} \overset{10}{=} \overline{\overset{\square}{\text{---}}} \overset{10}{=} \overline{\text{---}}$$

$$\text{So: } \overline{\overset{\square}{\text{---}}} = \overline{\text{---}}$$

(4) Strong complementarity



Special cases:

COPY: $m=0 \Rightarrow \text{---} \approx \text{---}$

$\text{---} \xrightarrow{\text{COPY}} \text{---}$

↑
10>

10...0>

Also: $\text{---} \xleftarrow{\text{SF}} \text{---} \xleftarrow{\text{TC}} \text{---} \xleftarrow{\text{SC}} \text{---} = \text{---}$

$\text{---} \xleftarrow{\text{SF}} \text{---} \xleftarrow{\text{TC}} \text{---} \xleftarrow{\text{SC}} \text{---} = \text{---}$

↑
11>

Discarding: $n=0 \Rightarrow \text{---} \approx \text{---}$

$\text{---} \xrightarrow{\text{XOR}} \text{---}$

↑
discard

"Applying a function,
Then throwing away the
output, is the same as
throwing away the inputs"

bialgebra $m=2, n=2 \Rightarrow \text{---} \approx \text{---}$

$= \boxed{\text{XOR}} - \boxed{\text{COPY}} = \boxed{\text{COPY}} - \boxed{\text{XOR}}$

$= \boxed{\text{COPY}} - \boxed{\text{XOR}}$

Rewriting examples

Thm (COMPLEMENTARITY)

□

$Z^x \approx \dots$

$\text{PF} \quad \dots = \quad \dots = \quad \dots = \quad \dots = \quad \dots$

$\approx \quad \dots \approx \quad \dots = \quad \dots \approx \quad \dots = \quad \dots \approx \quad \dots = \quad \dots \quad \boxed{\text{COPY}} \quad \boxed{\text{XOR}} = \quad \boxed{\text{discard}} \quad \boxed{\text{}} \quad \square$

$x \oplus x = 0$

Corollary:

$$\begin{array}{c} \text{id}_A \\ \text{id}_B \end{array} \approx \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array}$$

$$\begin{array}{c} \text{id}_A \\ \text{id}_B \end{array} \approx \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array}$$

$\text{CNOT} := \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array}$

$\begin{array}{c} a\pi \\ b\pi \end{array} \approx \begin{array}{c} a\pi \\ b\pi \end{array} \quad \frac{\text{TIC}}{\text{SC}} \quad \frac{\text{SF}}{\text{SF}} = \begin{array}{c} a\pi \\ (a \oplus b)\pi \end{array}$

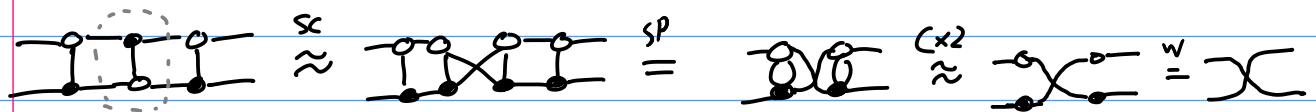
"controlled"

$$\langle \text{CNOT} | a, b \rangle = | a, a \oplus b \rangle$$

$$\begin{array}{c} \text{id}_A \\ \text{id}_B \end{array} \stackrel{\text{SF}}{=} \begin{array}{c} \text{id}_B \\ \text{id}_A \end{array} \stackrel{\text{com}}{=} \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array} \stackrel{\text{id}}{=} \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array}$$

$$S_v \left(\begin{array}{c} \text{id}_A \\ \text{id}_B \end{array} \right)^{-1} = \begin{array}{c} \text{id}_A \\ \text{id}_B \end{array}$$

Ex $3CNOT = SWAP$



Note: CNOTs are important:

Thm: any n-qubit unitary can be written as a circuit of $CNOT$, $Z[\alpha]$ and $X[\alpha]$ gates



So: $CNOT$ is the "only" 2-qubit interaction we need

Bell state

is an example of a **Product state**
 No connection

We also have **entangled states**

Bell state: $C = |00\rangle + |11\rangle$ The "cup"

Suppose $C = \begin{array}{c} \nearrow \\ \text{cup} \\ \searrow \end{array}$ for some $|\psi\rangle, |\phi\rangle$

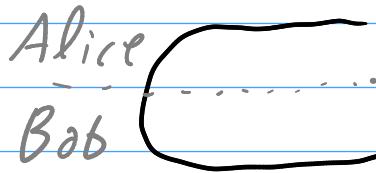
Then $= \begin{array}{c} \nearrow \\ \text{cup} \\ \searrow \end{array} = \begin{array}{c} \nearrow \\ \text{cup} \\ \searrow \end{array} = \begin{array}{c} \nearrow \\ |\psi\rangle \\ \searrow \end{array} \begin{array}{c} \nearrow \\ |\phi\rangle \\ \searrow \end{array}$

So everything would disconnect

Note = is known as the Yanking equation
 $(CAP \otimes ID) \circ (ID \otimes CUP) = ID$

Teleportation

1. Alice & Bob start with a Shared Bell state

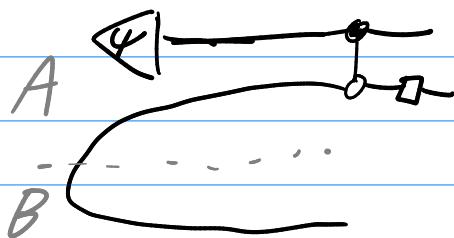


2. Then they may move far apart

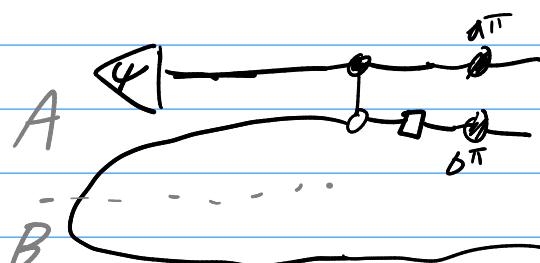
3. Alice picks a Q. state $|4\rangle$ she wants to send to Bob



4. Alice performs CNOT & Had on her states

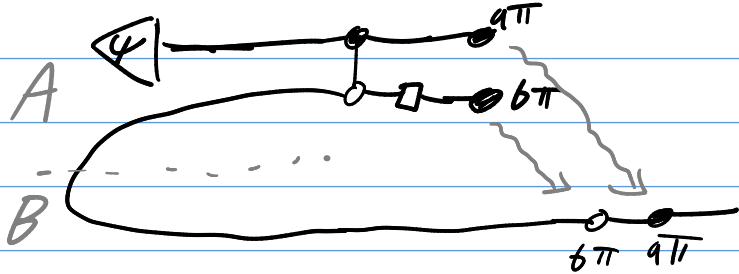


5. Then she measures both her qubits in $\{|0\rangle, |1\rangle\}$ basis, getting outcomes $a, b \in \{0, 1\}$

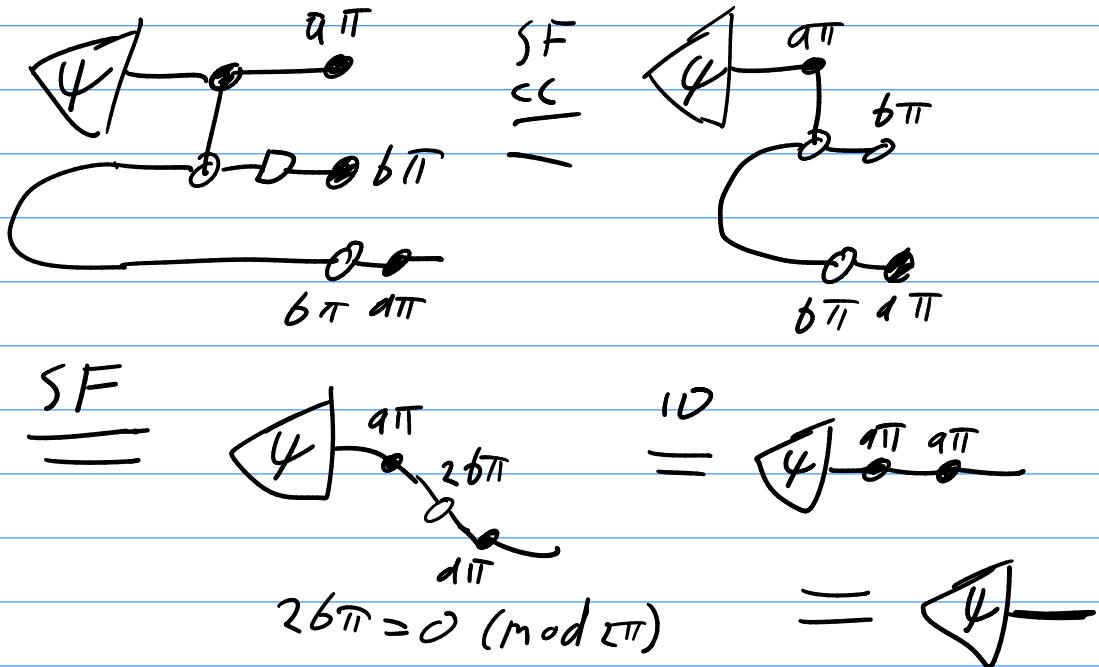


8

6. She communicates a, b to Bob, who performs a ~~$\frac{b\pi}{a}$~~ correction



7. Now Bob's state is Alice's former state:



Conclusion: Using Entanglement

& classical communication,

Alice can send Q .info to Bob

Needed! No FTL comm.

No cloning theorem

We can "clone", i.e. copy, classical states

$$\text{A} \rightarrow \text{C} = \begin{array}{c} \text{A} \\ \text{A} \end{array}$$

Is there some process $\text{A} \rightarrow \text{E}$ that
clones arbitrary Q. states?

Def: A map Δ is a **cloning process**

when

$$(1) \quad \text{A} \rightarrow \text{E} = \begin{array}{c} \text{A} \\ \text{A} \end{array} \quad \text{A normalised } \text{A}$$

$$(2) \quad \text{A} \rightarrow \text{B} \otimes \text{C} = \text{A} \rightarrow \text{E}$$

$$(3) \quad \text{A} \rightarrow \text{B} \otimes \text{C} = \begin{array}{c} \text{B} \\ \text{C} \end{array}$$

Thm: No Cloning Process exists

Pf: Suppose it did. Then $\text{C} \stackrel{(1)}{=} \text{B} \otimes \text{C}$

$$(2) \quad \text{C} \stackrel{\text{OCM}}{=} \text{B} \otimes \text{C} \stackrel{(3)}{=} \text{B} \otimes \text{C}$$

$$\text{OCM} \stackrel{?}{=} \text{C}. \text{ Then pick a state } \text{B} \otimes \text{C}: \quad \text{C} \stackrel{\text{OCM}}{=} \text{B} \otimes \text{C} \\ \Rightarrow \text{C} = \text{B} \otimes \text{C} \Leftrightarrow$$

Completeness

We have 5 types of rewrites:

$$\text{---} = \text{---}$$

$$\begin{array}{c} \alpha \\ \beta \end{array} = \alpha + \beta$$

$$\begin{array}{c} \alpha \\ \beta \end{array} = \alpha + \beta$$

$$\begin{array}{c} \alpha \\ \beta \end{array} = \begin{array}{c} \alpha \\ \beta \end{array}$$

$$\begin{array}{c} \alpha \\ \beta \end{array} \approx \begin{array}{c} \alpha \\ \beta \end{array}$$

$$\begin{array}{c} \alpha \\ \beta \end{array} = \begin{array}{c} \alpha \\ \beta \end{array}$$

Q: How much can we prove w/this?

Def: We say a set of rules is **complete**

When two diagrams representing the same linear map can always be rewritten into each other by the rules

Thm: The rules above are complete for diagrams with all phases in set

$\left\{ 0, \frac{\pi}{2}, \frac{\pi}{2}, -\frac{\pi}{2} \right\}$ The Clifford fragment

Thm: it is not complete over all phases, but adding one additional rule makes it complete

$$\begin{array}{c} \alpha \\ \beta \end{array} \approx \begin{array}{c} \alpha' \\ \beta' \\ \gamma' \end{array}$$

where $\alpha' = f_1(\alpha, \beta, \gamma)$ } complicated
 $\beta' = f_2(\alpha, \beta, \gamma)$ } functions
 $\gamma' = f_3(\alpha, \beta, \gamma)$

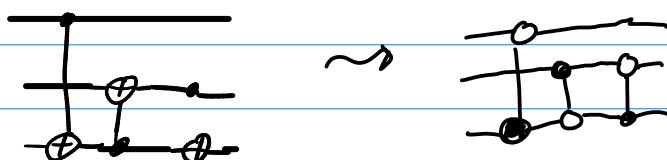
CNOT Circuits & Phase Free ZX Diagrams

CIRCUITS MADE JUST OUT OF $\begin{array}{c} \text{I} \\ \otimes \end{array}$

$$\begin{array}{c} \text{I} \\ \otimes \end{array} = \begin{array}{c} \text{I} \\ \otimes \end{array}$$

ZX-DIAGS MADE OUT OF $\begin{array}{c} \text{I} \\ \otimes \end{array}$ AND $\begin{array}{c} \text{X} \\ \otimes \end{array}$

Prop Any CNOT circuit is equal to a phase free ZX-diagram.



Q: What about the converse?

Thm: (Unitary) phase-free ZX-diags \rightsquigarrow CNOT circuits.

Parities

$$\text{CNOT } |x, y\rangle \mapsto |x, x \oplus y\rangle$$

$$\text{CNOT } |x, y\rangle \mapsto |f_1(x, y), f_2(x, y)\rangle \quad \text{where} \quad \begin{cases} f_1(x, y) = x \\ f_2(x, y) = x \oplus y \end{cases}$$

Def A function of the form $f(x_1, \dots, x_n) = x_{i_1} \oplus \dots \oplus x_{i_k}$ is called a **parity map**.

Def The field \mathbb{F}_2 has elements $\{0, 1\}$ where:

$$x \cdot y := x \wedge y \quad x + y = x \oplus y \quad (\text{i.e. } x + y \bmod 2)$$

Sometimes we call some $x \in \mathbb{F}_2$ a parity.

$$\text{par}(\vec{b}) = \sum_i b_i \quad \text{in } \mathbb{F}_2$$

$\text{par}(\vec{b}) = 0$ means \vec{b} has an even # of 1's
 $\text{par}(\vec{b}) = 1$ means odd #.

Parities for subsets of bits:

$$(1 \ 0 \ 1 \ 1) \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = b_1 \oplus b_3 \oplus b_4$$

Multiple parities at once:

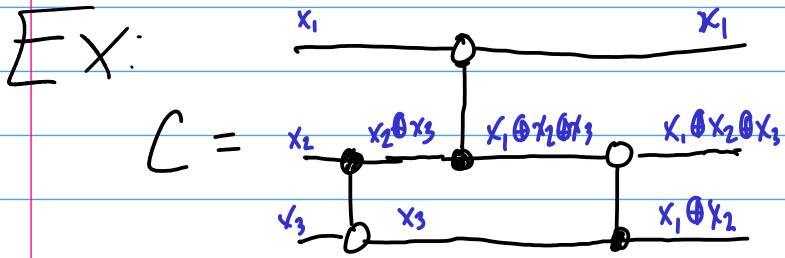
$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{parity matrix.}} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} b_1 \oplus b_3 \oplus b_4 \\ b_2 \oplus b_3 \\ b_1 \oplus b_4 \\ b_4 \end{pmatrix}$$

The action of a CNOT circuit on basis elements is defined by an invertible parity matrix:

$$C(b_1, \dots, b_n) = |c_1, \dots, c_n\rangle$$

where $P \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$. $P P^{-1} = ID$

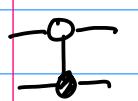
Note: P is $n \times n$, so not exponentially large



$$C|x_1, x_2, x_3\rangle = |x_1, x_1 \otimes x_2 \otimes x_3, x_1 \otimes x_2\rangle$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_P \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_1 \otimes x_2 \otimes x_3 \\ x_1 \otimes x_2 \end{pmatrix}$$

Special case: Single CNOT.



$$|x, y\rangle \mapsto |x, x \otimes y\rangle$$

$$\underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_P \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ x \otimes y \end{pmatrix}$$

More generally:

$$j \leftrightarrow \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} = E^{ij}$$

elementary matrix

$$E^{ji} A = A'$$

row $j = \text{row } j + \text{row } i$

$$A E^{ji} = A'$$

col $j = \text{col } i + \text{col } j$

Suppose $P E^{ij_1} \dots E^{i_k j_k} = I$,

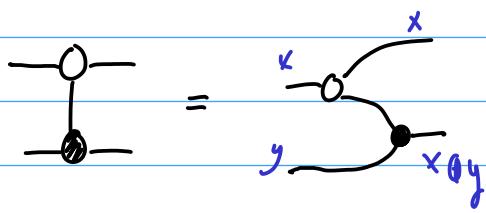
then $P = E^{i_k j_k} \dots E^{i_1 j_1}$

\uparrow \nwarrow \nearrow
parity matrix CNOT gates!

Algorithm: CNOT-SYNTH:

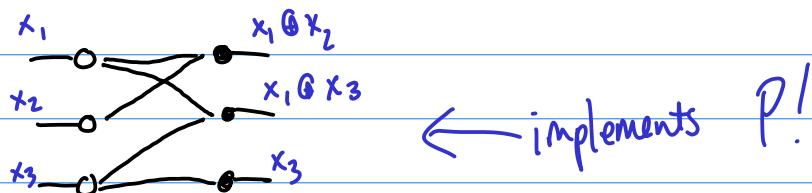
- * Start w/ Parity matrix P , empty circ. C .
- * Do Gauss-Jordan reduction of columns of P .
 - whenever an elem. col operation E^{j_i} is applied,
append CNOT_{ji} to C .
- * C now implements P .

Parity maps in ZX

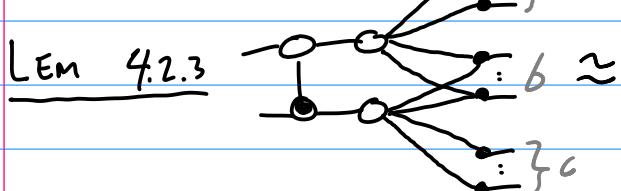
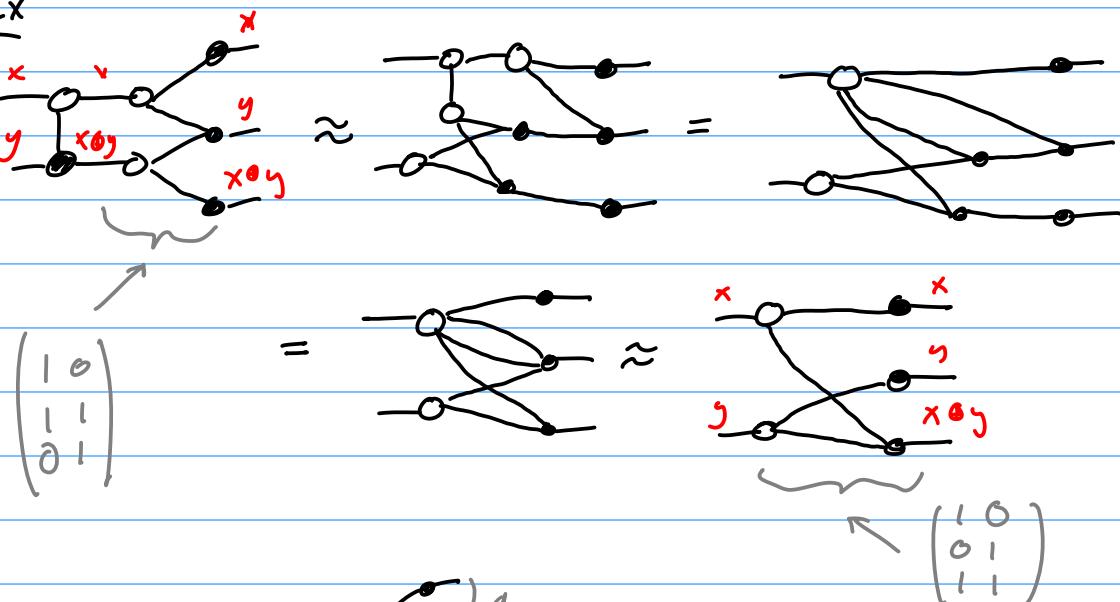


More general parity maps:

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_1 \otimes x_2 \\ x_1 \otimes x_3 \\ x_2 \otimes x_3 \end{pmatrix}$$



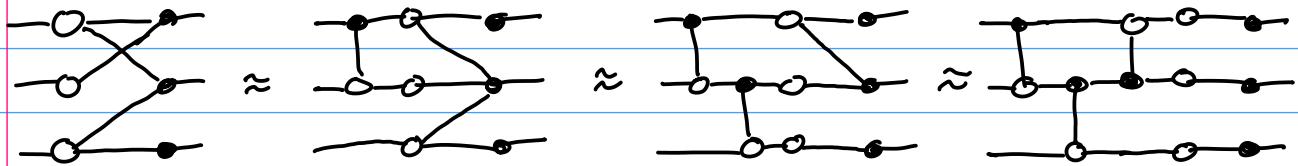
Ex



$$\begin{array}{l} a \{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ b \{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ c \{ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \quad \begin{array}{l} c_1 = c_1 + c_2 \\ \longleftarrow \\ c_2 = c_1 + c_2 \end{array} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Ex

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{C_2=C_2+C_1} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{C_3=C_3+C_2} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{C_1=C_1+C_2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

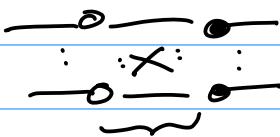


Def A spider is called

- * an input spider if it is conn. to an input
- * an output spider --- output
- * an interior spider otherwise.

Def A phase-free ZX-diagram is in parity normal form

- every Z spider is conn. to exactly 1 input
- every X --- output
- no wires between spiders of the same type
- no parallel wires



P parity matrix.

Thm: A ZX-diagram in parity NF
can be rewritten into a CNOT circuit

Missing Piece: From arbitrary Phase-free
ZX-diagram to Parity NF