Lecture 10

Application 3 Completeness of the ZX-calculus for Clifford diagrams.

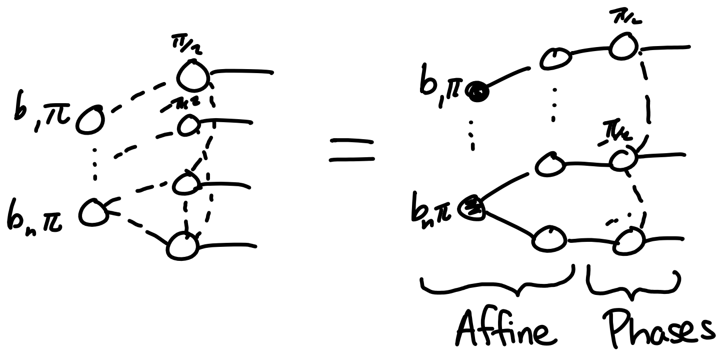
Thm (Completeness) For Clifford ZX-diagrams D_1, D_2, if $D_1 = D_2$ then $D_1 \overset{ZX}{=} D_2$.

matrices are equal

can (efficiently!) transform D_1 to D_2 with the ZX-calc.

Idea: Look at the AP form.

Def A graph-like ZX-diagram is in AP-form if all interior spiders:
- have phase $\in 0, \pi$
- are only connected to boundary spiders.

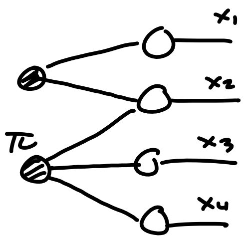


$$= \sum_{\vec{x},\, A\vec{x}=\vec{b}} e^{i\frac{\pi}{2}\cdot\phi} |\vec{x}\rangle$$

$\mathcal{A} = \{\vec{x} \mid A\vec{x} = \vec{b}\}$ is an affine subspace of $\mathbb{F}_2^n$.

:= a solution to a set of linear eqns, e.g:

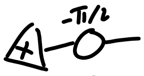
$$\mathcal{A} = \left\{ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \,\middle|\, \begin{matrix} x_1 \oplus x_2 = 0 \\ x_2 \oplus x_3 \oplus x_4 = 1 \end{matrix} \right\} \Longleftrightarrow$$

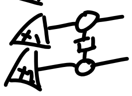
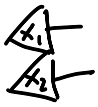


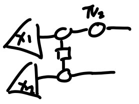
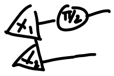
$$\approx \sum_{\vec{x},\, A\vec{x}=\vec{b}} |\vec{x}\rangle$$

ϕ is a phase polynomial

 $= e^{i\pi \cdot (\frac{1}{2} x)} |x\rangle$

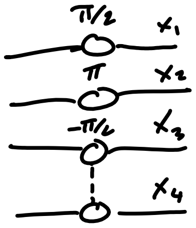
 $= e^{i\pi (-\frac{1}{2} x)} |x\rangle$

 $= (-1)^{x_1 x_2} \cdot$  $= e^{i\pi (x_1 x_2)}$ 

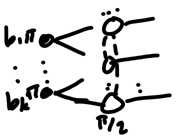
 $= e^{i\pi (x_1 x_2)} \cdot$  $= e^{i\pi (x_1 x_2)} e^{i\pi (\frac{1}{2} x_1)}$  $= e^{i\pi (x_1 x_2 + \frac{1}{2} x_1)}$ 

phase polynomial

$$U|\vec{x}\rangle = e^{i\pi \cdot \phi} |\vec{x}\rangle \text{ where } \phi = \tfrac{1}{2}x_1 - \tfrac{1}{2}x_3 + x_2 + x_3 x_4$$

$U =$ 

Def An AP-form is in reduced AP-form if it is $\emptyset$ or A is in reduced echelon form and the polynomial ϕ only contains free variables from A.

 $= \sum_{\vec{x}, A\vec{x} = \vec{b}} e^{i\pi\phi} |\vec{x}\rangle$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

free var x_3

echelon form

Prop Reduced AP-form is unique.

Pf (Linear algebra)

Prop For Clifford diag D, $D \overset{ZX}{=} D'$ ← reduced AP-form.

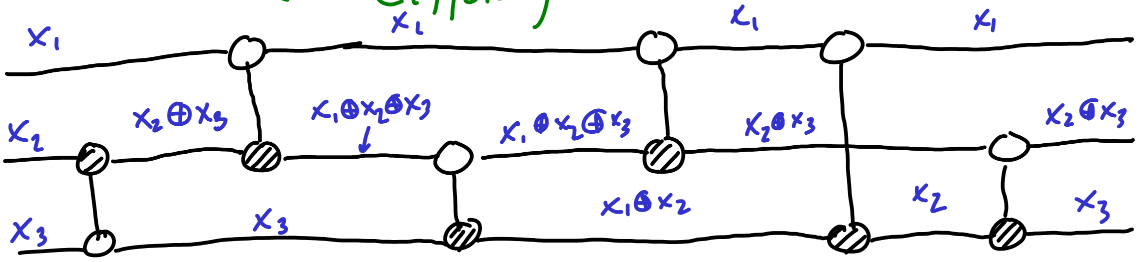
Pf (ZX can do Gaussian elimination!)

Cor Completeness!

CNOT + phase circuits

(non-Clifford)

$U =$

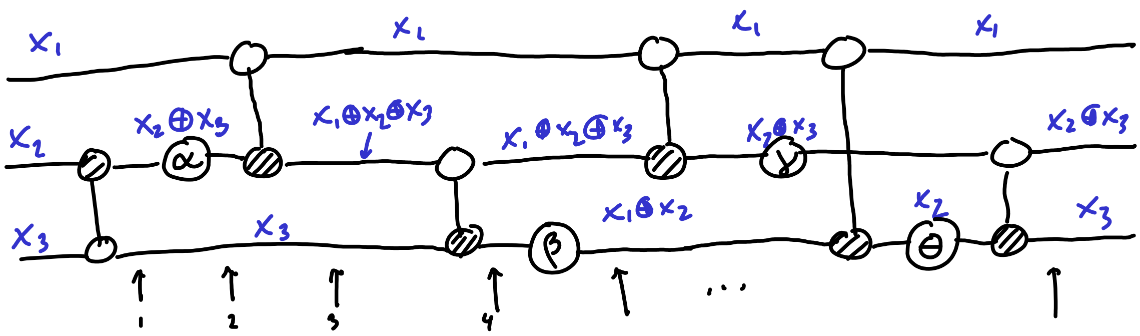


$$U|x_1 x_2 x_3\rangle = |x_1, x_2 \oplus x_3, x_3\rangle$$

Q: What happens when we add phase gates?

$$Z[\alpha] :: |x\rangle \mapsto e^{i\alpha \cdot x}|x\rangle$$

$V =$



$$|x_1 x_2 x_3\rangle \overset{1}{\longmapsto} |x_1, x_2 \oplus x_3, x_3\rangle$$

$$\overset{2}{\longmapsto} e^{i\alpha \cdot (x_2 \oplus x_3)} |x_1, x_2 \oplus x_3, x_3\rangle$$

$$\overset{3}{\longmapsto} e^{i\alpha (x_2 \oplus x_3)} |x_1, x_1 \oplus x_2 \oplus x_3, x_3\rangle$$

$$\overset{4}{\longmapsto} e^{i\alpha \cdot (x_2 \oplus x_3)} |x_1, x_1 \oplus x_2 \oplus x_3, x_1 \oplus x_2\rangle$$

$$\longmapsto e^{i[\alpha \cdot (x_2 \oplus x_3) + \beta \cdot (x_1 \oplus x_2)]} |x_1, x_1 \oplus x_2 \oplus x_3, x_1 \oplus x_2\rangle$$

$$\longmapsto \ldots$$

$$\longmapsto e^{i[\alpha \cdot (x_2 \oplus x_3) + \beta \cdot (x_1 \oplus x_2) + \gamma \cdot (x_2 \oplus x_3) + \theta \cdot x_2]} |x_1, x_2 \oplus x_3, x_3\rangle$$

Prop Any CNOT+phase circuit describes a unitary of the form:

$$U :: |\vec{x}\rangle \mapsto e^{i\phi(\vec{x})} |L\vec{x}\rangle.$$

phase polynomial parity matrix.

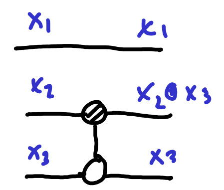
From the example above: $L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ and

$$\phi(x_1, x_2, x_3) = (\alpha + \gamma) \cdot (x_2 \oplus x_3) + \beta \cdot (x_1 \oplus x_2) + \theta \cdot x_2$$

phase-folding

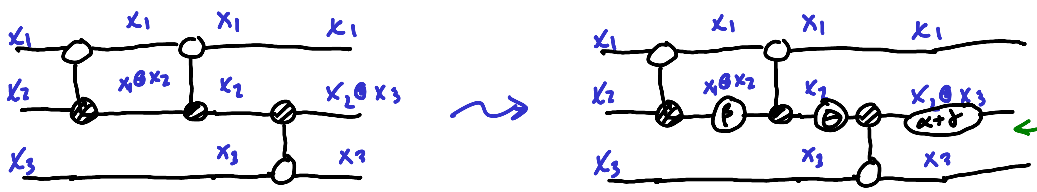
Q: can we re-synthesise a circuit for (L, ϕ)?

For L, we have:



To get ϕ, we need to place Z-phases on wires labelled:
$x_2 \oplus x_3$, $x_1 \oplus x_2$, and x_2

Only $x_1 \oplus x_2$ is missing, so lets (temporarily) create it:

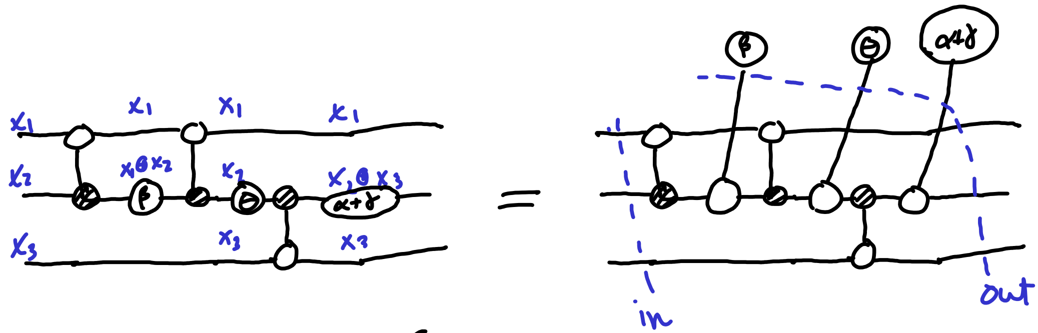


phase-folding

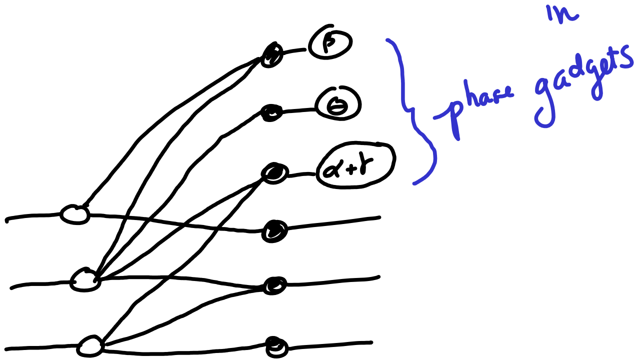
Lecture 11

Phase polynomials, graphically (aka. phase gadgets)

Ex



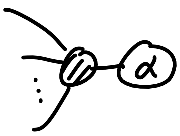
$\overset{\text{phase-free simp.}}{=}$



1-legged:

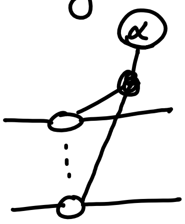
$$:: \ |x\rangle \longmapsto \begin{cases} 1 & \text{if } x=0 \\ e^{i\alpha} & \text{if } x=1 \end{cases} = e^{i\alpha \cdot x}$$

k-legged phase gadget:

$\sqrt{2}^{(k-1)}$ 

$$:: \ |x_1 .. x_k\rangle \longmapsto e^{i\alpha \cdot (x_1 \oplus .. \oplus x_k)}$$

In a diagonal unitary:

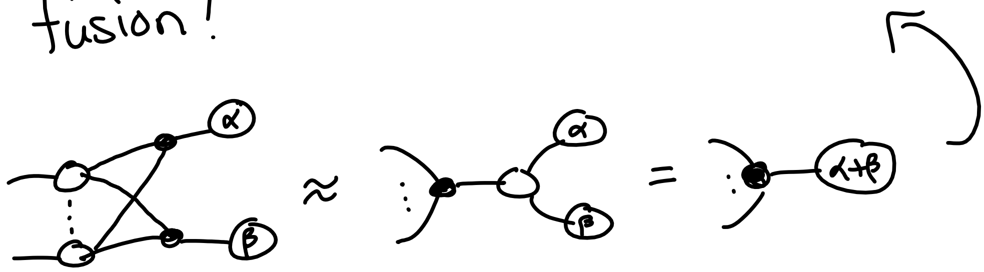
$\sqrt{2}^{(k-1)}$ 

$$:: \ |x_1 .. x_k\rangle \longmapsto e^{i\alpha \cdot (x_1 .. x_k)} \, |x_1 .. x_k\rangle$$

Q: What happens when there is phase folding?

identical connectivity. fuse using (sc)+(sp).

A: Gadget fusion!



Algorithm: CNOT + phase optimisation.

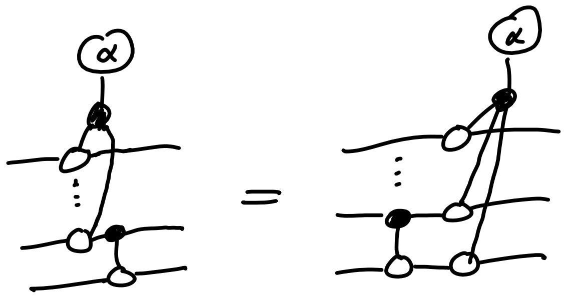
1. unfuse phases and treat as outputs.
2. Compute PNF of phase-free part.
3. perform gadget fusion (* and other phase-poly reductions!)

?? → 4. extract a CNOT + phase circuit.

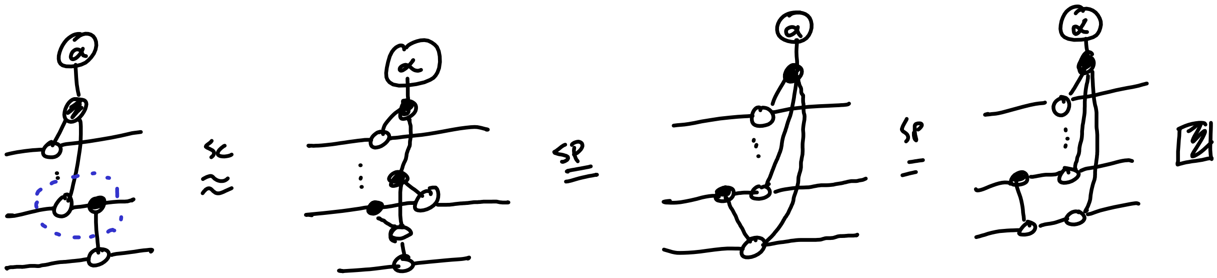
There are choices for step 4.

Naïve approach: "CNOT ladders"

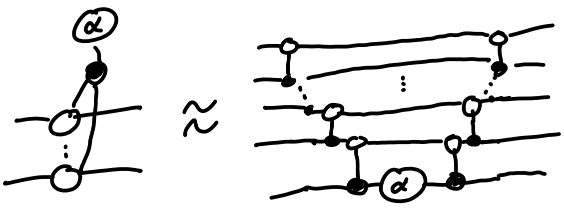
Prop



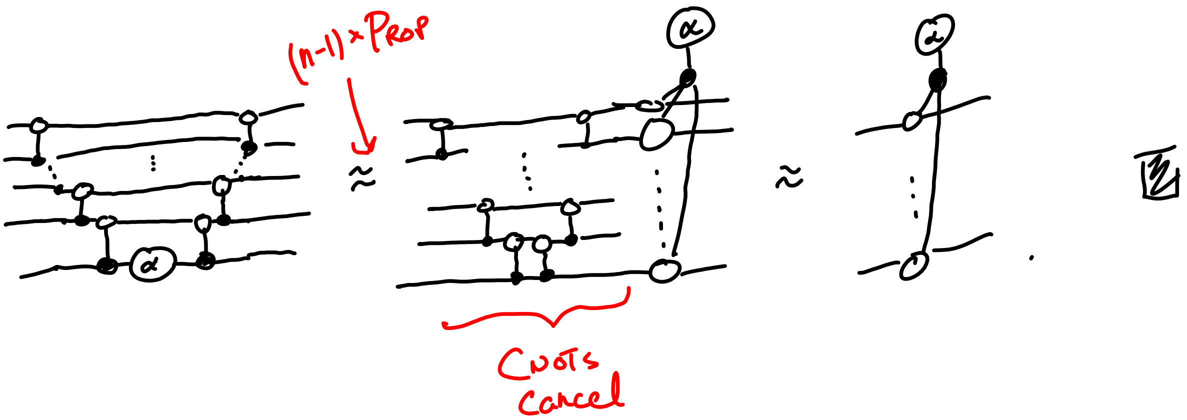
Pf



CoR

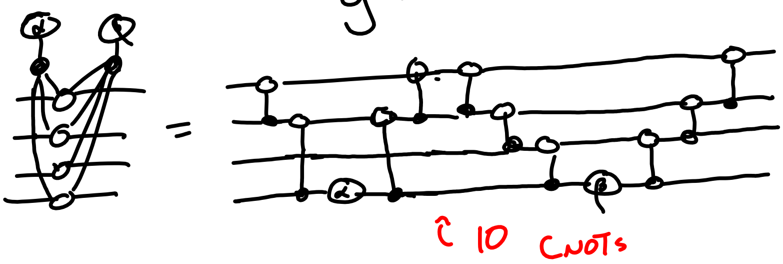


Pf

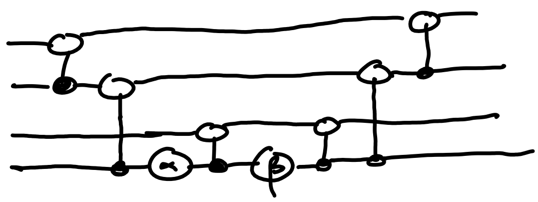


Naïve extraction : 1. unfuse a phase gadget & replace using CoR 1.
2. repeat until no phase gadgets
3. synthesise CNOT circuit from phase-free diag.

* Lots of wasted CNOT gates! e.g.



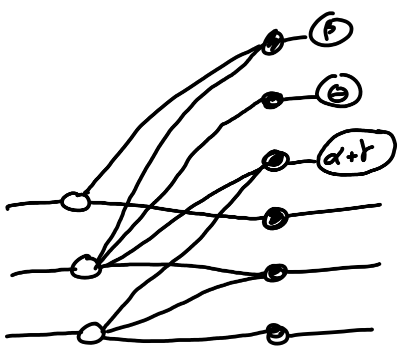
Vs.



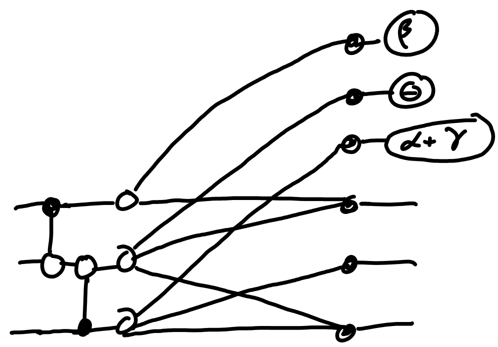
6 CNOTs

"T-par" style extraction [Amy, Maslov, Mosca 2013]

1. write an "extended biadjacency matrix"
2. identify a set of k linearly independent rows
3. reduce each row to a unit vector with column ops.
4. "extract phases" and repeat.



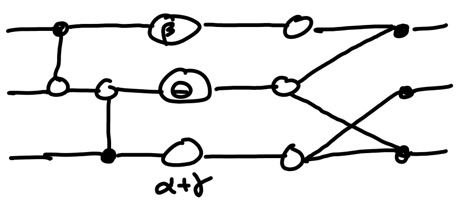
$\approx$ " " $\approx$



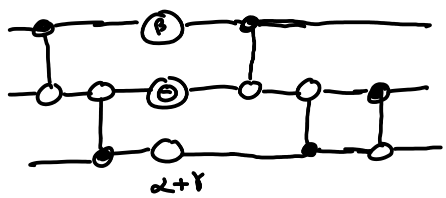
$\overset{SP}{=}$ $*$

$$\begin{array}{l} \text{gadgets} \\ \\ \\ \text{outputs} \\ \\ \\ \end{array} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ \hdashline 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{C_2 = C_2 + C_1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{C_2 = C_2 + C_3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hdashline 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$* =$



$\approx$



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

PRO's: · very good at low non-Clifford depth (i.e. layers of non-Cliff gates).

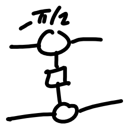
· gets better with ancillae!

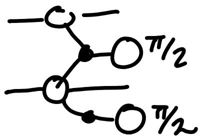
CONS · CNOT count/depth is inconsistent.

* Better for CNOT count: Gray-synth [Amy, Azimzadeh, Mosca 2017]

Lecture 12 High-level gates.

We've seen 2 kinds of phase polynomials:

"Multilinear" form, e.g.  $:: |x,y\rangle \mapsto e^{i\pi\cdot(-\frac{1}{2}x + x\cdot y)} |x,y\rangle$

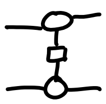
"XOR"/ Phase-gadget form, e.g.  $:: |x,y\rangle \mapsto e^{i\pi\cdot(x\oplus y + y)}$

These two forms are related:

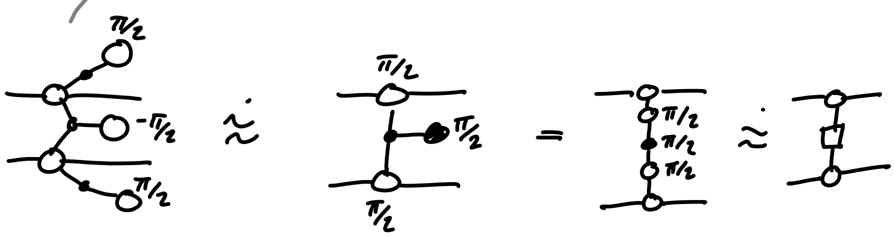
$$\overset{\text{XOR}}{x \oplus y} = x \overset{\text{plus}}{+} y \underbrace{- 2xy}_{\text{"correction"}} \qquad (x,y \in \{0,1\})$$

$$-2xy = x\oplus y - x - y$$

$$\Rightarrow xy = \tfrac{1}{2}(x + y - x\oplus y)$$

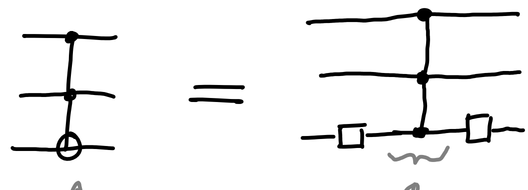
 $:: |xy\rangle \mapsto e^{i\pi\cdot(xy)} |xy\rangle$

$$= e^{i\pi(\frac{1}{2}x + \frac{1}{2}y - \frac{1}{2}x\oplus y)} |xy\rangle$$



Some gates are easy to write in multilinear form.

Consider:

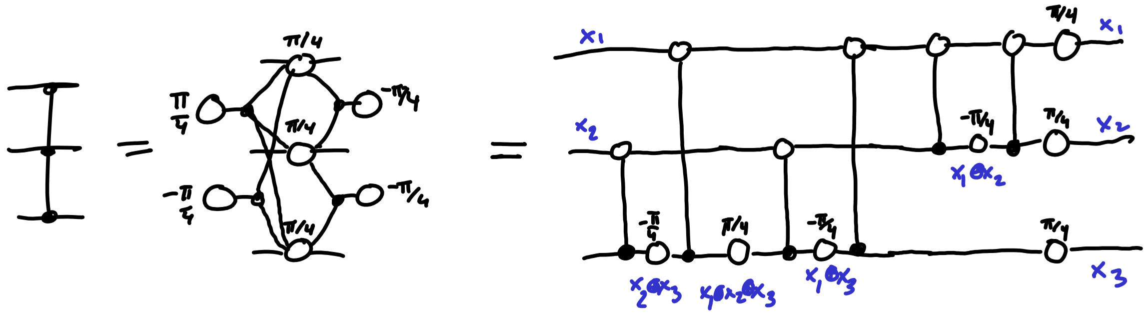


Tof not diagonal   CCZ diagonal

$$CCZ|x_1x_2x_3\rangle = \begin{cases} |x_1x_2x_3\rangle & \text{if } x_1\cdot x_2 = 0 \\ |x_1x_2\rangle \otimes \underbrace{Z|x_3\rangle}_{(-1)^{x_3}|x_3\rangle} & \text{if } x_1\cdot x_2 = 1 \end{cases}$$

$$= (-1)^{x_1x_2x_3}|x_1x_2x_3\rangle = e^{i\pi\cdot x_1x_2x_3}|x_1x_2x_3\rangle$$

$$\begin{aligned} x_1(x_2x_3) &= \tfrac{1}{2}x_1\cdot(x_2+x_3-x_2\oplus x_3) \\ &= \tfrac{1}{2}(x_1x_2+x_1x_3-x_1(x_2\oplus x_3)) \\ &= \tfrac{1}{4}(x_1+x_2-x_1\oplus x_2+\cancel{x_1}+x_3-x_1\oplus x_3-\cancel{x_1}-x_2\oplus x_3+x_1\oplus x_2\oplus x_3) \\ &= \tfrac{1}{4}(x_1+x_2+x_3-x_1\oplus x_2-x_1\oplus x_3-x_2\oplus x_3+x_1\oplus x_2\oplus x_3) \end{aligned}$$



(see Nielsen + Chuang p.182)

Exercise: Why does N+C end up with an extra S gate?

Translation of CCZ into XOR form is a special case of discrete Fourier transform.

Prop For any function $\phi : \mathbb{F}_2^n \to \mathbb{R}$,

dot product, i.e. parities.

$$\phi(\vec{x}) = \frac{-1}{2^{n-1}} \sum_{\vec{y}}' \tilde{\alpha}_{\vec{y}} (\vec{x}\cdot\vec{y})$$

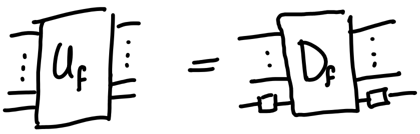
where $\tilde{\alpha}_{\vec{y}} = \frac{-1}{2^{n-1}} \sum_{\vec{z}} (-1)^{\vec{y}\cdot\vec{z}} \cdot \phi(\vec{y})$ are the Fourier coefficients.

In the CCZ case, taking the Fourier xform of $\phi(\vec{x}) = \begin{cases} 1 & \text{if } x_1x_2x_3 = 1 \\ \emptyset & \text{o.w.} \end{cases}$

gives: $\begin{cases} \tilde{\alpha}_{100} = \tilde{\alpha}_{010} = \tilde{\alpha}_{001} = \frac{1}{4} \\ \tilde{\alpha}_{110} = \tilde{\alpha}_{101} = \tilde{\alpha}_{011} = -\frac{1}{4} \\ \tilde{\alpha}_{111} = \frac{1}{4}. \end{cases}$

This gives a general strategy for synthesising classical oracles.

1. Write:



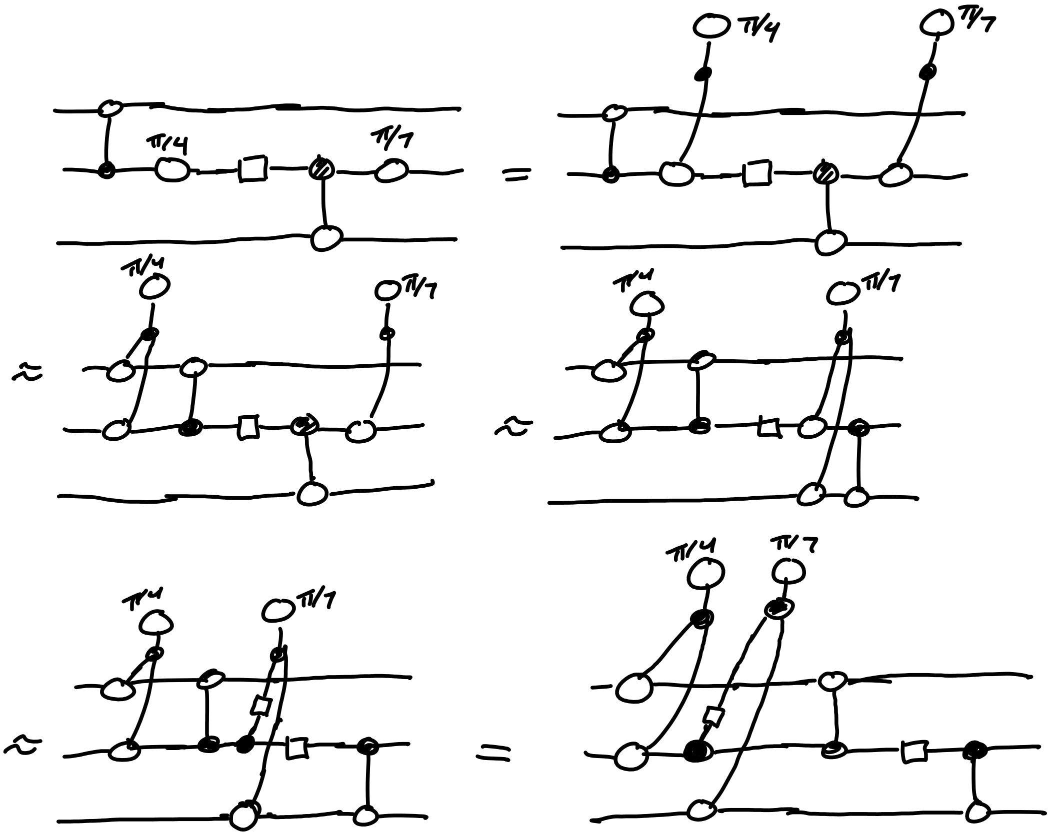
$$D_f|\vec{x}, y\rangle := e^{i\pi\cdot\phi}|\vec{x}, y\rangle \quad \text{where } \phi(\vec{x}, y) = f(\vec{x})\cdot y$$

2. Compute Fourier coeffs of ϕ.
3. Synthesise D_f as CNOT+Phase circuit.

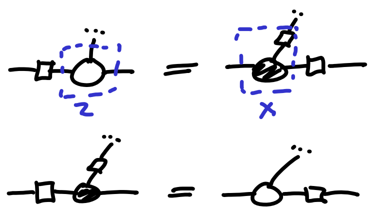
Pauli Gadgets

Clifford + Phase is a universal family.

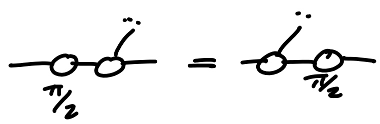
Q: Can we move all the non-Clifford phases out?

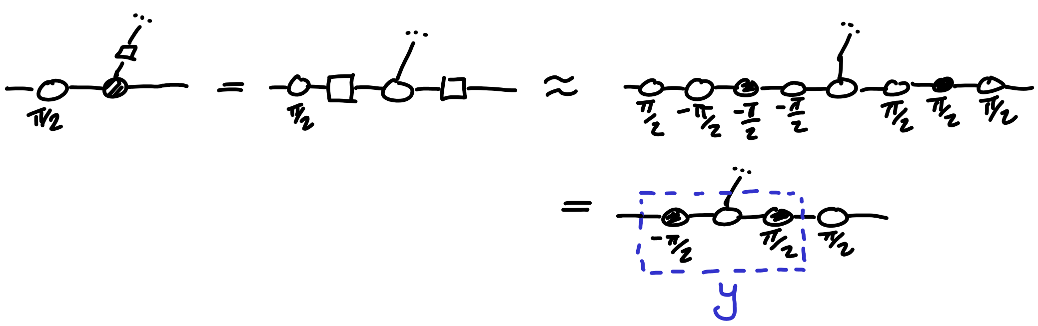


H gates:

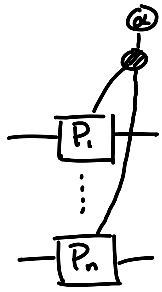


S gates:

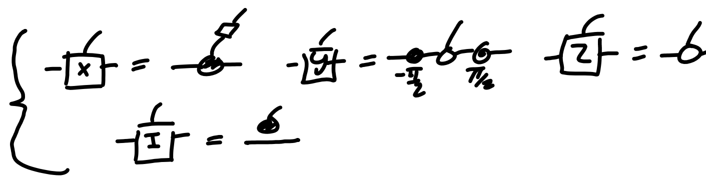




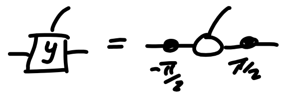
Prop For $\vec{P} = P_1 \otimes .. \otimes P_n$ with $P_i \in \{I, X, Y, Z\}$ the map:

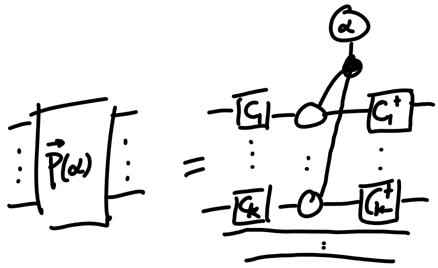


where:



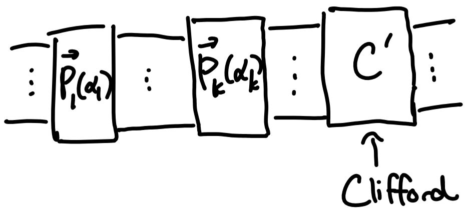
is unitary. It is called the Pauli gadget $\vec{P}(\alpha)$.

Pf Note  and . So



for Cliff. unitaries C_i. Since phase gadgets are unitary, so is $\vec{P}(\alpha)$. ☒

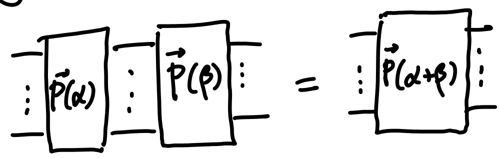
Thm Any Clifford+phase circuit can be written as:

$C =$ 

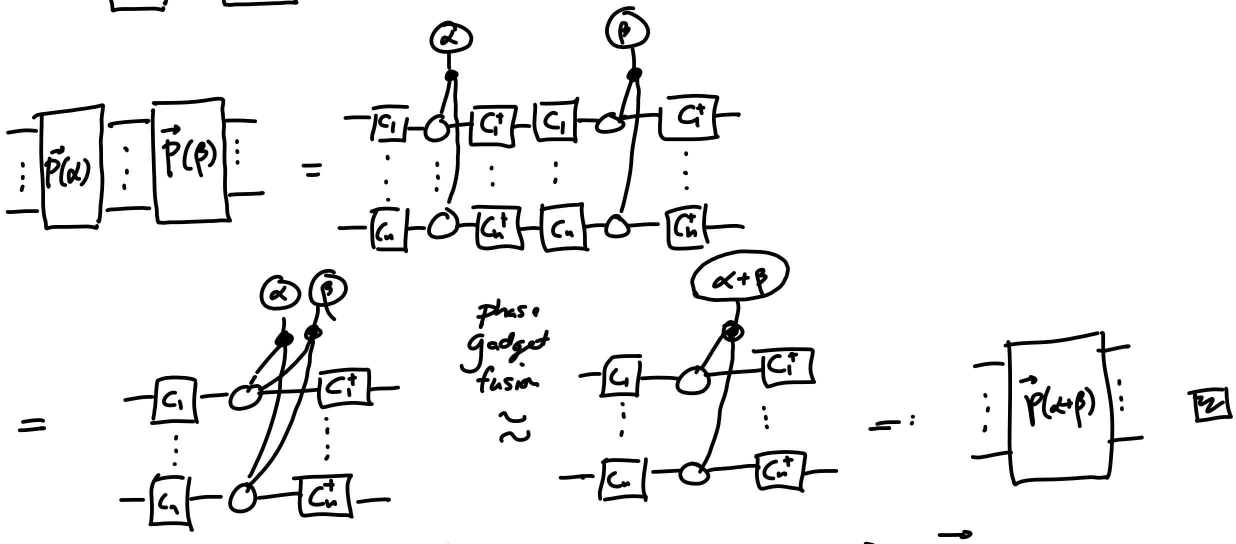
Pf (Idea) • Show Pauli gadgets commute past all Clifford gates.

• Move phases out of C, one at a time. ☒

Prop (Pauli gadget fusion.)



Pf



Prop For Paulis $\vec{P}, \vec{Q}$ if $\vec{P}\vec{Q} = \vec{Q}\vec{P}$, then $\vec{P}(\alpha)\vec{Q}(\beta) = \vec{Q}(\beta)\vec{P}(\alpha)$.

Pf Exercise/book (Hint: it's complementarity!)

Algorithm Pauli "phase folding".

1. Compute Pauli gadget form of a circuit.
2. Commute PG's and combine phases where possible.
3. Merge PG's with Clifford phases into the Clifford part.
4. Repeat until no more reductions.
5. Extract circuit.*

* like with CNOT+phase, there are many options.