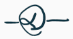
Introduction to Quantum Programming and Semantics

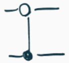
Louis Lemonnier

THE UNIVERSITY of EDINBURGH
informatics

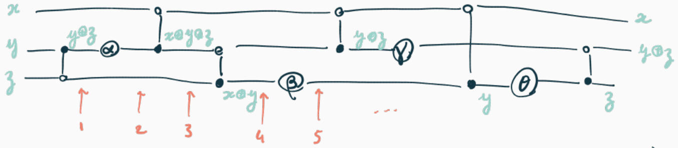
10th March 2026. Lecture 14: Circuit optimisation with ZX-calculus

Phase polynomials

 $: |x\rangle \longmapsto e^{i\alpha}|x\rangle$

 $: |x, y\rangle \longmapsto |x, x\oplus y\rangle$

$$U: |\vec{x}\rangle \longmapsto e^{i\underbrace{P(\vec{x})}_{\text{phase polynomial}}}|\underbrace{L}_{\text{parity matrix}}\vec{x}\rangle$$



$$|x, y, z\rangle \xmapsto[1]{} |x, y\oplus z, z\rangle \xmapsto[2]{} e^{i\alpha(y\oplus z)}|x, y\oplus z, z\rangle \xmapsto[3]{} e^{i\alpha(y\oplus z)}|x, x\oplus y\oplus z, z\rangle$$

$$\xmapsto[4]{} e^{i\alpha(y\oplus z)}|x, x\oplus y\oplus z, x\oplus y\rangle \xmapsto[5]{} e^{i(\alpha(y\oplus z)+\beta(x\oplus y))}|x, x\oplus y\oplus z, x\oplus y\rangle$$

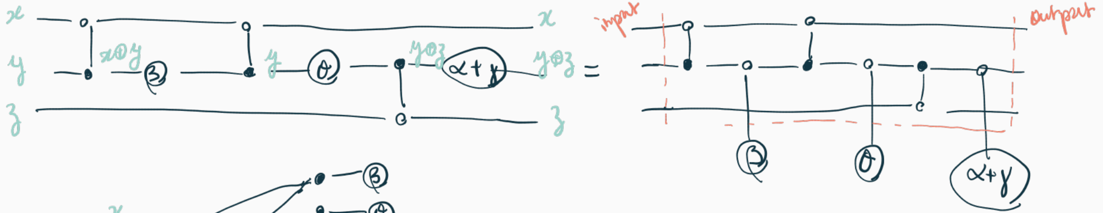
$$\cdots \longmapsto e^{i(\alpha(y\oplus z)+\beta(x\oplus y)+\gamma(y\oplus z)+\theta y)}|x, y\oplus z, z\rangle$$

Goal

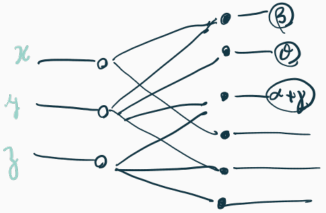
- Identify the graphical structure of phase polynomials
- Use ZX calculus to rewrite/optimise the diagrams
- Identify how to extract a circuit

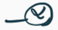
Phase gadgets

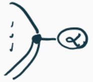
Phase gadgets



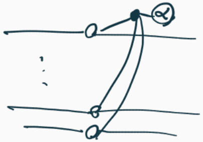
=



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

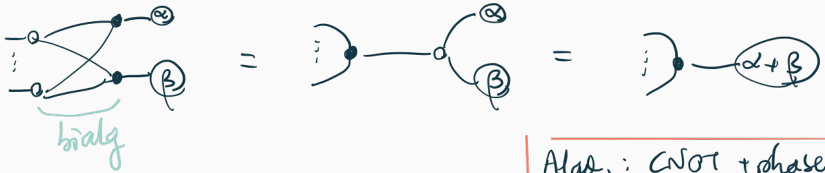
k-legs :  : $|x_1 \cdots x_k\rangle \mapsto e^{i\alpha(x_1 \oplus \cdots \oplus x_k)}$

diagonal unitary matrix :

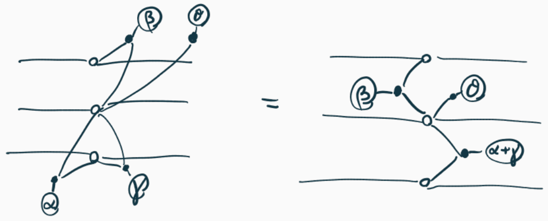


: $|x_1 \dots x_k\rangle \mapsto e^{i\alpha(x_1 \oplus \cdots \oplus x_k)} |x_1 \cdots x_k\rangle$

Phase gadgets can fuse!



e.g.



Algo.: CNOT + phases opti.

1. unfuse phases, treat them as output
2. phase-free → CNOTs
3. perform gadget fusion
4. extract CNOT + phase circuit

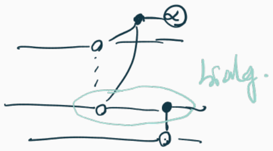
(several choices possible for 4.)

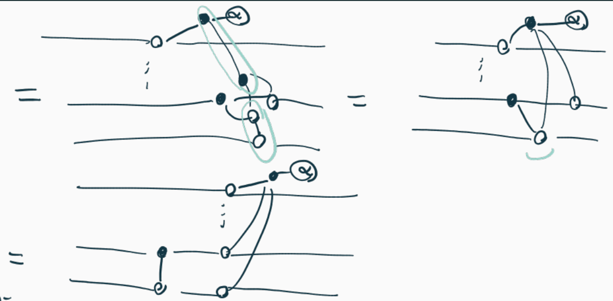
extraction!

Circuit ~~optimisation~~

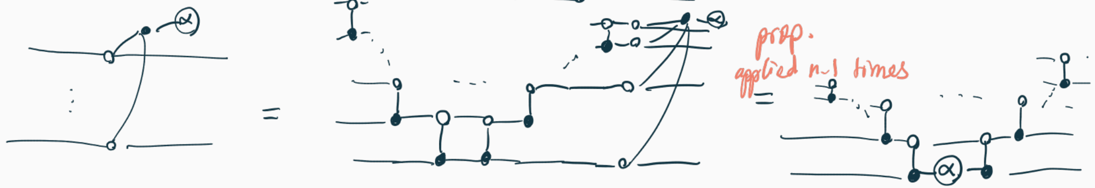
CNOT ladders

prop.





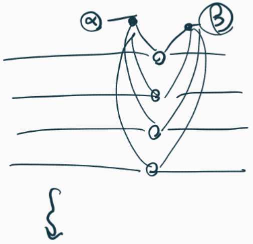
cor.

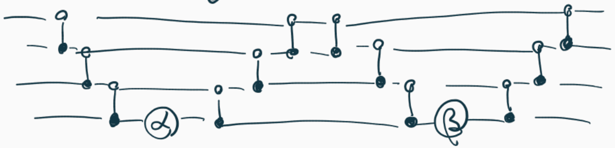


CNOT ladders

Naive extraction algo.

1. unfuse gadgets, then cor.
2. repeat (until can't)
3. Synthesise the CNOTs

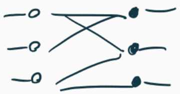




12 CNOT gates.

CNOT ladders

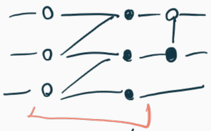
Adjacency matrix.



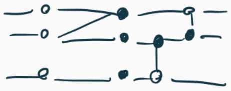
$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Gauss-Jordan reduct°

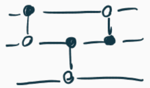
$$\xrightarrow{(2) = (2)+(1)} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \xrightarrow{(2) = (2)+(3)} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



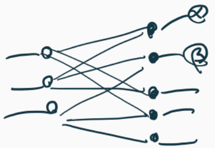
=



=



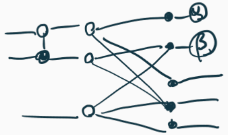
Same story with phase gadgets:



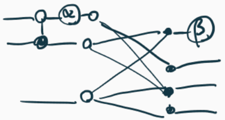
$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hdashline 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

columns

$$\xrightarrow{(2) = (2)+(1)} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ \hdashline 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$



$\rightsquigarrow$



Summary

Better extraction algo.

1. Compute adjacency matrix
2. Gauss - Jordan reduction
3. Extract phase
4. repeat.

- Phase polynomials and gadgets bring structure
- Turn a circuit into ZX-diagram, simplify, and extract