Introduction to Quantum Programming and Semantics

Lecture 14: Circuit optimisation

Chris Heunen

University of Edinburgh

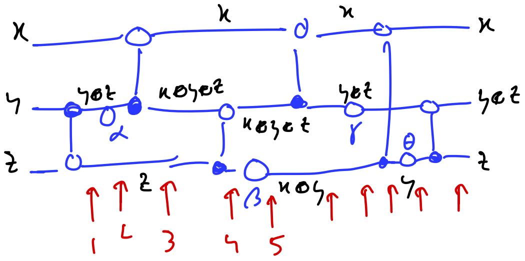
Overview

- Universal quantum circuits
- Phase polynomials
- Phase gadgets
- Circuit opimisation

Phase polynomials

Phase polynomials

What happens if we add (arbitrary) phases to CNOT/Clifford circuits



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

$$|xyz\rangle \xmapsto{1} |x, y\oplus z, z\rangle$$

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

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

$$\xmapsto{4} e^{i\alpha(y\oplus z)}|x, x\oplus y\oplus z, x\oplus y\rangle$$

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

$$\mapsto \dots$$

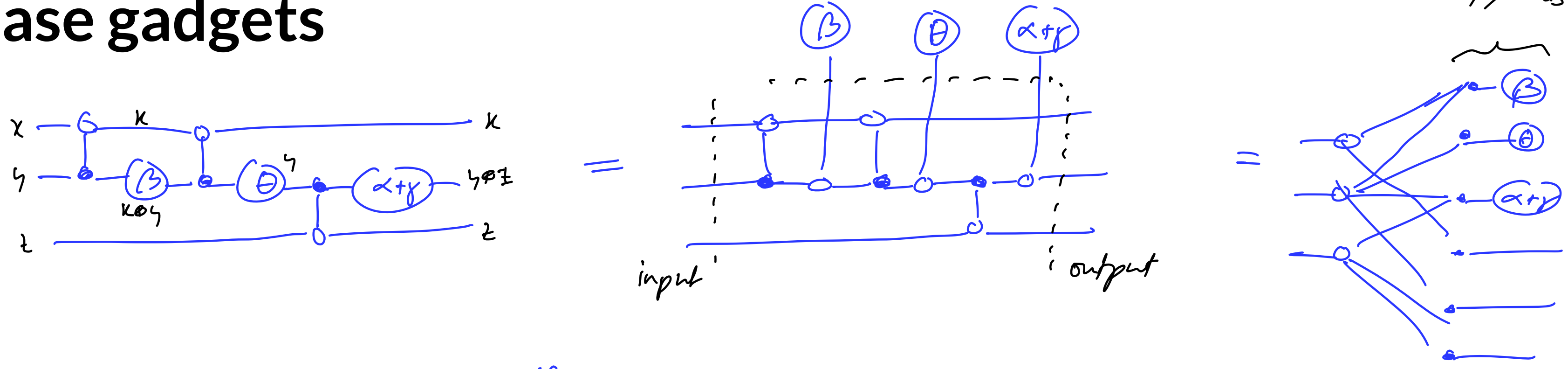
$$\mapsto e^{\underbrace{i[\alpha(y\oplus z)+\beta(x\oplus y)+\gamma(y\oplus z)+\theta y]}_{\text{"phase polynomial"}}}|x, y\oplus z, z\rangle$$

any CNOT+phase circuit describes a unitary of the form

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

Phase gadgets

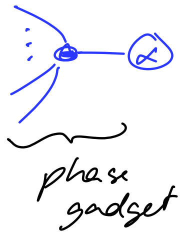
Phase gadgets



1-legs:

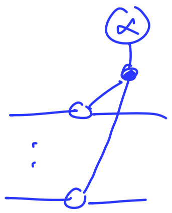
$$: \quad |x\rangle \mapsto \begin{cases} 1 & \text{if } x=0 \\ e^{i\alpha} & \text{if } x=1 \end{cases} = e^{i\alpha\cdot x}$$

k-legs:



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

as a diagonal unitary:

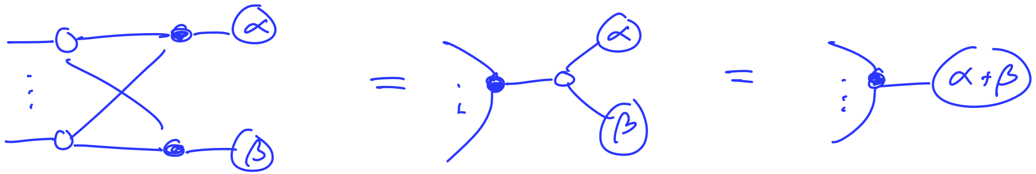


$$: \quad |x_1 \cdots x_k\rangle \mapsto e^{i\alpha(x_1 \cdots x_k)} |x_1 \cdots x_k\rangle$$

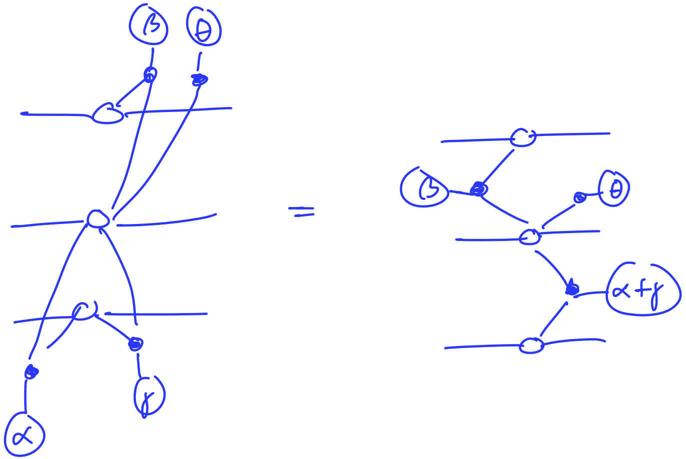
Phase gadgets

can fuse!

rule:



e.g.:



algorithm for CNOT+phase optimisations

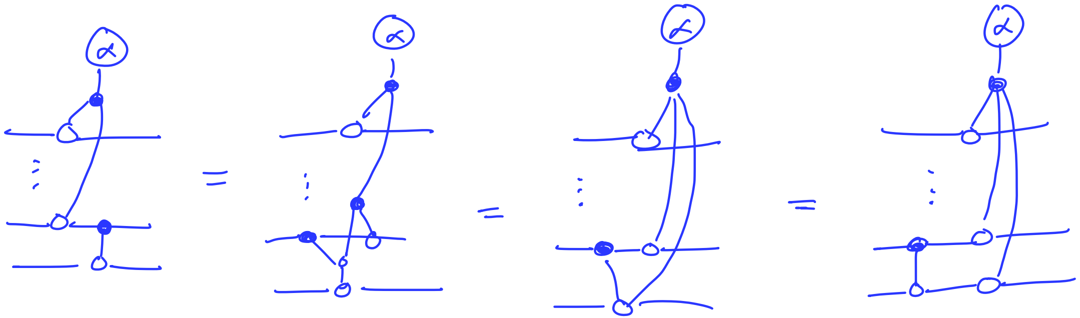
1. unfuse all phases and treat as outputs
2. compute parity normal form of phase-free part
3. perform gadget-fusion
4. extract CNOT+phase circuit.

(there are several choices for step 4.)

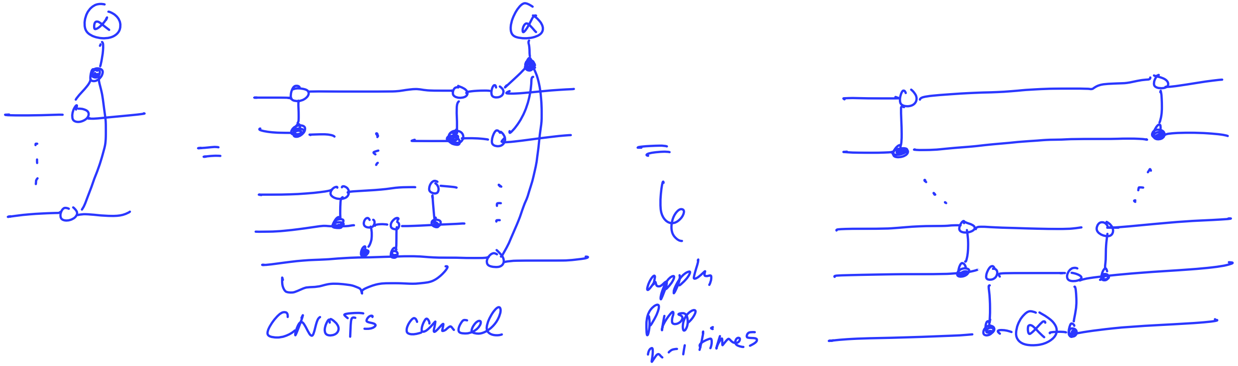
Circuit optimisation

CNOT ladders

Prop:



Cor:

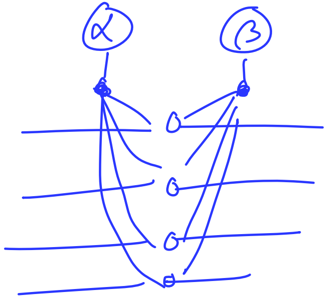


Naive extraction:

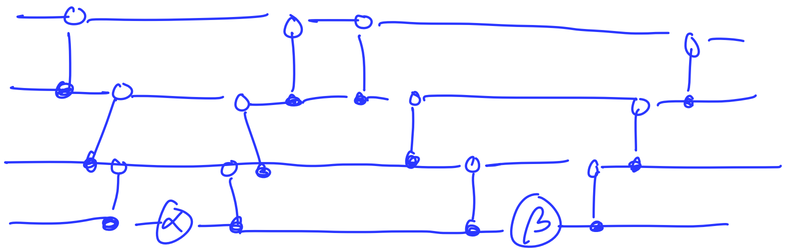
1. unfuse a phase gadget + replace using Cor
2. repeat until can't
3. synthesise CNOT circuit from phase-free diagram

CNOT ladders

lots of wasted gates

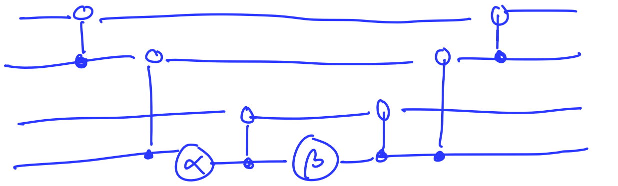


=



12 CNOTS

vs

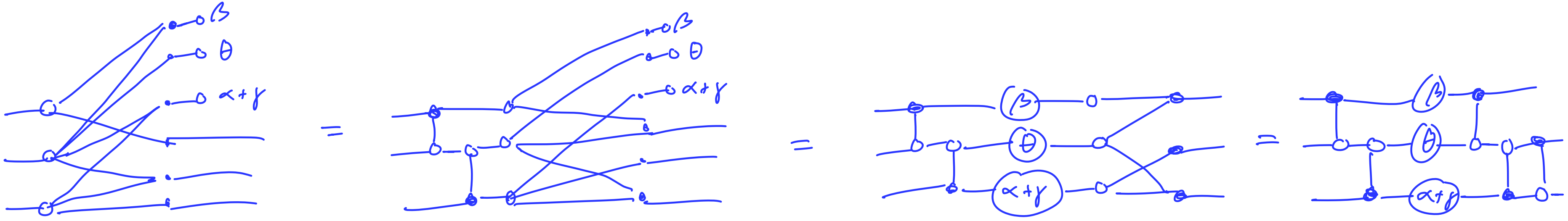


6 CNOTS

CNOT ladders

better extraction strategy:

1. find "extended biadjacency matrix"
2. identify k lin. indep. rows
3. reduce each row
4. "extract" phases and repeat



$$\begin{array}{c} \text{gadgets}\left\{\begin{array}{c} \\ \\ \\ \end{array}\right. \\ \text{outputs}\left\{\begin{array}{c} \\ \\ \\ \end{array}\right. \end{array} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

add col 1 to col 2

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

add col3 to col2

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

pro: - good at low Clifford depth
- better with ancillas

con: CNOT depth inconsistent

Summary:

- ZX diagrams with pi/4 phases fully universal but difficult
- Phase polynomials/gadgets bring structure
- Can be used to rewrite circuits