Introduction to Quantum Programming and Semantics

Lecture 11: Classical quantum circuits

Chris Heunen

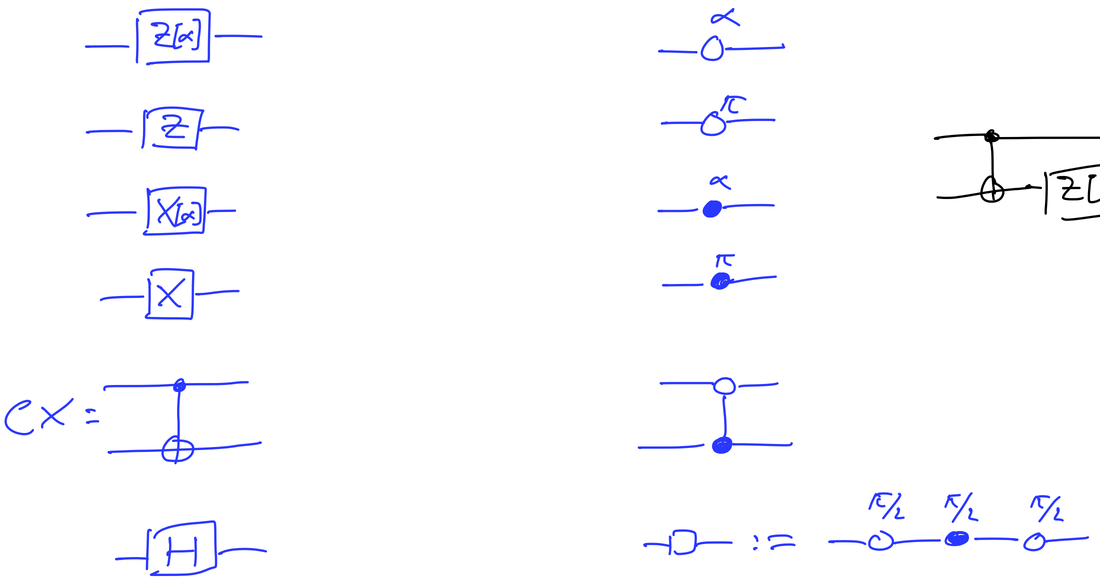
University of Edinburgh

Overview

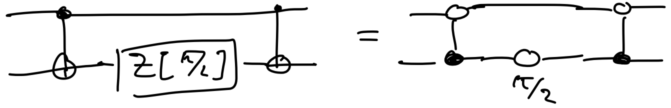
- CNOT circuits
- Bennett’s trick
- Phase-free ZX diagrams

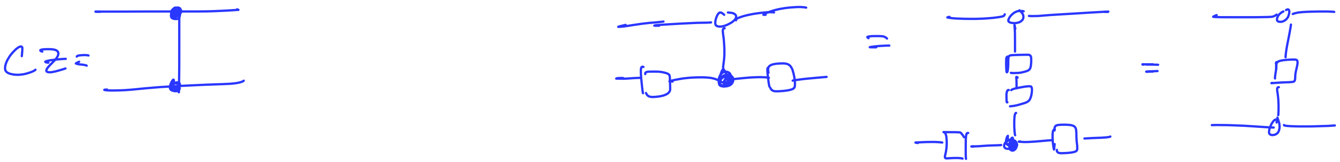
dictionary

circuits ⟶ ZX-diagrams

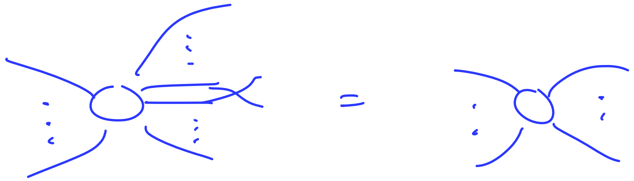


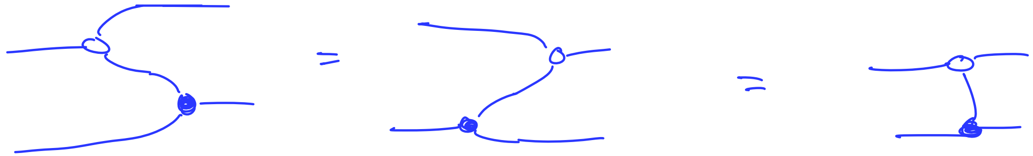
e.g.:





ZX diagrams have extreme 'only connectivity matters':





ZX calculus is tool to reason about circuits. Solve problems like:

- synthesis: given high-level description of computation/unitary, find circuit that implements it

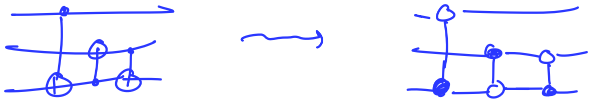
- optimisation: given a circuit C that implements U, find a smaller C' that also implements U

- (classical) simulation: given C that implements U, (not given quantum computer) and an input $|\psi\rangle$,
  - compute measurement probabilities for $U|\psi\rangle$ (strong)
  - sample measurement outcome for $U|\psi\rangle$ (weak)

CNOT circuits

Parity

any CNOT circuit is a phase-free ZX diagram



what about converse?

def: a function of form $f(x_1,\dots,x_n) = x_{i_1} \oplus \cdots \oplus x_{i_n}$ is called a *parity map*

can think about linear algebra over $\mathbb{F}_2 = \{0,1\}$,

$x \cdot y = x$ AND y

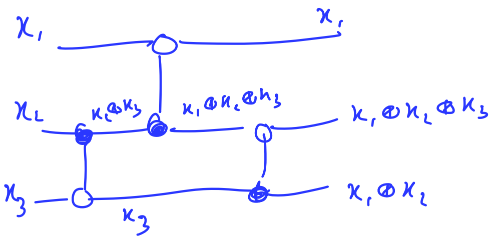
$x + y = x$ XOR y

e.g. $$\begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = a \oplus c \oplus d$$

$$\underbrace{\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{parity matrix}} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \oplus c \oplus d \\ b \oplus c \\ a \oplus d \\ d \end{pmatrix}$$

Parity matrix

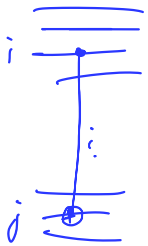
: can turn CNOT circuit into parity matrix.

$C =$ 

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

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

more generally:



gives parity matrix

$$\begin{array}{cc} \begin{array}{c} i \\ \downarrow \end{array} & \\ \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ 1 & & & \ddots \; 1 \end{pmatrix} & \leftarrow j \end{array} = E^{ij} \text{ elementary matrix}$$

$E^{ij} A = A' \Leftarrow$ row$'j$ = row j + row i

$A E^{ij} = A' \Leftarrow$ col$'j$ = col j + col i

CNOT circuit synthesis

- start with a parity matrix P, empty circuit C
- do Gauss-Jordan reduction on columns of P
  whenever E^{ij} applied, append $CNOT^{ji}$ to C
- repeat until eliminated P

C implements P

Bennett’s trick

Reversible computing

where do circuits come from?

one source: classical computations

$f: \{0,1\}^n \to \{0,1\}^n \quad \leadsto \quad (\mathbb{C}^2)^{\otimes n} \xrightarrow{U_f} (\mathbb{C}^2)^{\otimes n}$

$|x\rangle \longmapsto |f(x)\rangle$

$f: \{0,1\}^n \to \{0,1\} \quad \underset{\text{Bennett's trick}}{\leadsto} \quad (\mathbb{C}^2)^{\otimes n+1} \xrightarrow{U_f} (\mathbb{C}^2)^{\otimes n+1}$

$|x,y\rangle \longmapsto |x, f(x) \oplus y\rangle$

e.g.: $\text{NOT}: \{0,1\} \to \{0,1\} \quad \leadsto \quad X: \begin{array}{l} |0\rangle \mapsto |1\rangle \\ |1\rangle \mapsto |0\rangle \end{array}$

$\text{CNOT}: \{0,1\}^2 \to \{0,1\}^2 \quad \leadsto \quad CX: |x,y\rangle \longmapsto |x, x \text{ XOR } y\rangle$

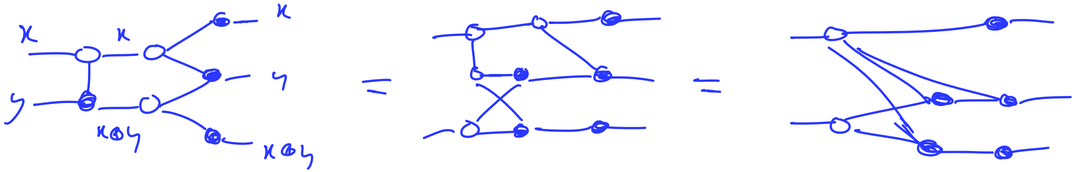
$= |x, x \oplus y\rangle$

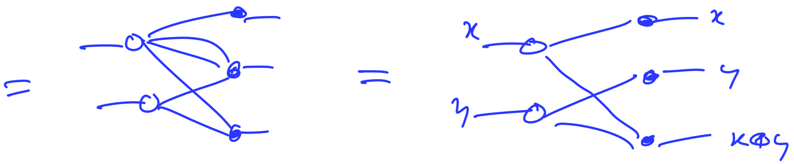
Phase-free ZX diagrams

Parity normal form

parity matrix

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





parity normal form

Reduction to parity form

unitary phase-free ZX-diagram $\rightsquigarrow$ parity normal form $\rightsquigarrow$ parity matrix $\rightsquigarrow$ CNOT circuit

unitary phase-free ZX-diagram $\longleftarrow$ dictionary $\longleftarrow$ CNOT circuit

Summary:

- ZX diagrams are (complexity-theoretically) hard to rewrite algorithmically
- Special class of CNOT diagrams easier, comes down to parity checking
- Bennett’s trick makes classical functions reversible, useful for oracles
- Special class of phase-free ZX diagrams easier