

# Introduction to Quantum Computing

Miriam Backens (they/them)

Inria & Loria

`miriam.backens@inria.fr`

“New trends in Computing” summer school, Strasbourg, 2024

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Bits and Boolean functions

- ▶ Data is stored as bits:

$$b \in \{0, 1\}$$

# Bits and Boolean functions

- ▶ Data is stored as **bits**:

$$b \in \{0, 1\}$$

- ▶ Sequences of bits form **bit strings**:

$$\mathbf{b} \in \{0, 1\}^n$$

# Bits and Boolean functions

- ▶ Data is stored as **bits**:

$$b \in \{0, 1\}$$

- ▶ Sequences of bits form **bit strings**:






$$\mathbf{b} \in \{0, 1\}^n$$

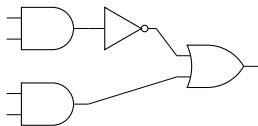
- ▶ Transformations are given by **Boolean functions**:

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^m$$

# Logic circuits

Cannot directly implement most Boolean functions, instead decompose them as **circuits** over a small set of **logic gates**.






name	symbol	function
AND		$x, y \mapsto x \wedge y$
OR		$x, y \mapsto x \vee y$
NOT		$x \mapsto \neg x$
XOR		$x, y \mapsto x \oplus y$
COPY		$x \mapsto x, x$

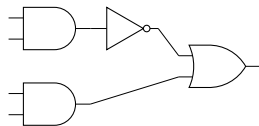


$$f(x, y, z, w) = \neg(x \wedge y) \vee (z \wedge w)$$

# Logic circuits

Cannot directly implement most Boolean functions, instead decompose them as **circuits** over a small set of **logic gates**.

name	symbol	function
AND		$x, y \mapsto x \wedge y$
OR		$x, y \mapsto x \vee y$
NOT		$x \mapsto \neg x$
XOR		$x, y \mapsto x \oplus y$
COPY		$x \mapsto x, x$



$$f(x, y, z, w) = \neg(x \wedge y) \vee (z \wedge w)$$

Number of gates in circuit can be used as measure of computational complexity.



## Linear algebra notation for bits

Associate to each bit value a vector, which can also be written as a **ket** in so-called Dirac notation:

$$0 \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix} =: |0\rangle$$

$$1 \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix} =: |1\rangle$$

## Linear algebra notation for bits

Associate to each bit value a vector, which can also be written as a **ket** in so-called Dirac notation:

$$0 \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix} =: |0\rangle \qquad 1 \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix} =: |1\rangle$$

The vector associated to a bit string is the **tensor product** (Kronecker product) of the bit vectors – essentially a unary encoding:

$$01 \mapsto |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} =: |01\rangle$$

## Linear algebra notation for bits

Associate to each bit value a vector, which can also be written as a **ket** in so-called Dirac notation:

$$0 \mapsto \begin{pmatrix} 1 \\ 0 \end{pmatrix} =: |0\rangle \qquad 1 \mapsto \begin{pmatrix} 0 \\ 1 \end{pmatrix} =: |1\rangle$$

The vector associated to a bit string is the **tensor product** (Kronecker product) of the bit vectors – essentially a unary encoding:

$$01 \mapsto |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} =: |01\rangle$$

In Dirac notation, the  $\otimes$  symbol is sometimes left out, e.g.  $|0\rangle |1\rangle = |0\rangle \otimes |1\rangle$ .

# Linear algebra notation for Boolean functions

Transformations are represented as matrices, e.g. the AND gate becomes

$$\begin{array}{c} 0 \\ 1 \end{array} \begin{array}{cccc} & 00 & 01 & 10 & 11 \\ \left( \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right)$$

## Linear algebra notation for Boolean functions

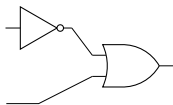
Transformations are represented as matrices, e.g. the AND gate becomes

$$\begin{array}{c} 00 \quad 01 \quad 10 \quad 11 \\ \begin{array}{c} 0 \\ 1 \end{array} \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

Then  $0 \wedge 1$  can be computed as

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} |01\rangle = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle$$

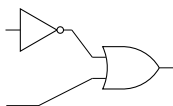
## Linear algebra for logic circuits



Use tensor product for parallel composition and matrix product for serial composition:

$$\text{OR} (\text{NOT} \otimes I)$$

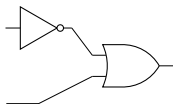
## Linear algebra for logic circuits



Use tensor product for parallel composition and matrix product for serial composition:

$$\text{OR}(\text{NOT} \otimes I) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \left( \left( \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \right)$$

## Linear algebra for logic circuits

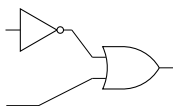


Use tensor product for parallel composition and matrix product for serial composition:

$$\begin{aligned}\text{OR}(\text{NOT} \otimes I) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \left( \left( \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \right) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \left( \begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right)\end{aligned}$$



## Linear algebra for logic circuits



Use tensor product for parallel composition and matrix product for serial composition:

$$\begin{aligned} \text{OR}(\text{NOT} \otimes I) &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \left( \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \left( \begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \end{aligned}$$

## Reversible logic gates

A logic gate is **reversible** if the corresponding matrix is invertible.

- ▶ The AND gate is not reversible since  $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  is not square.

## Reversible logic gates

A logic gate is **reversible** if the corresponding matrix is invertible.

- ▶ The AND gate is not reversible since  $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  is not square.
- ▶ The NOT gate is reversible since it corresponds to the matrix

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

## Reversible logic gates

A logic gate is **reversible** if the corresponding matrix is invertible.

- ▶ The AND gate is not reversible since  $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  is not square.
- ▶ The NOT gate is reversible since it corresponds to the matrix

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

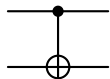
## Reversible logic gates

A logic gate is **reversible** if the corresponding matrix is invertible.

- ▶ The AND gate is not reversible since  $\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  is not square.
- ▶ The NOT gate is reversible since it corresponds to the matrix

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

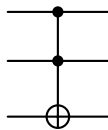
The CNOT gate is considered a reversible version of XOR:  $x, y \mapsto x, y \oplus x$



$$\begin{matrix} & 00 & 01 & 10 & 11 \\ 00 & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ 01 & \\ 10 & \\ 11 & \end{matrix}$$

# The Toffoli gate and functional universality

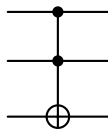
$$x, y, z \mapsto x, y, z \oplus (x \wedge y)$$



	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	1	0

# The Toffoli gate and functional universality

$$x, y, z \mapsto x, y, z \oplus (x \wedge y)$$



	000	001	010	011	100	101	110	111
000	1	0	0	0	0	0	0	0
001	0	1	0	0	0	0	0	0
010	0	0	1	0	0	0	0	0
011	0	0	0	1	0	0	0	0
100	0	0	0	0	1	0	0	0
101	0	0	0	0	0	1	0	0
110	0	0	0	0	0	0	0	1
111	0	0	0	0	0	0	1	0

The Toffoli gate is **functionally universal**: any Boolean function can be computed by a logic circuit consisting of Toffoli gates (and constant inputs).

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions



# The quantum bit, or qubit

The state of a qubit is described by a normalised vector in the Hilbert space  $\mathbb{C}^2$ :

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle \qquad \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$$

# The quantum bit, or qubit

The state of a qubit is described by a normalised vector in the Hilbert space  $\mathbb{C}^2$ :

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle \quad \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$$

The values  $\alpha, \beta$  are called **amplitudes**. If  $\alpha, \beta$  are both non-zero, the state is a **superposition** of  $|0\rangle$  and  $|1\rangle$ .

# The quantum bit, or qubit

The state of a qubit is described by a normalised vector in the Hilbert space  $\mathbb{C}^2$ :

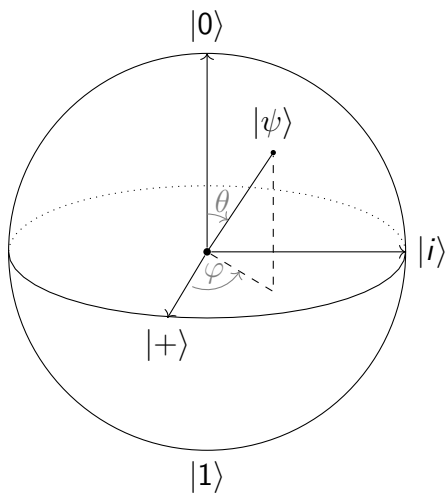
$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle \quad \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$$

The values  $\alpha, \beta$  are called **amplitudes**. If  $\alpha, \beta$  are both non-zero, the state is a **superposition** of  $|0\rangle$  and  $|1\rangle$ .

It is physically impossible to distinguish two states that differ only by a global factor  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $|\gamma| = 1$ . Single-qubit states can thus be written as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right) |1\rangle \quad \theta \in [0, \pi], \varphi \in [0, 2\pi)$$

## Visualising single-qubit states on the Bloch sphere



$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\varphi} \sin\left(\frac{\theta}{2}\right) |1\rangle$$

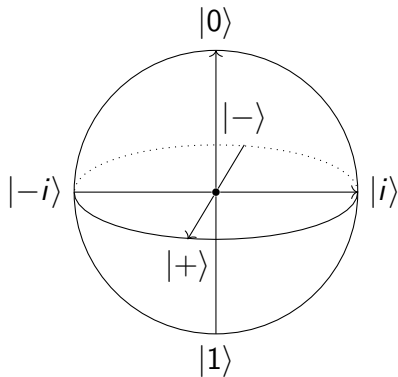
where  $\theta \in [0, \pi]$ ,  $\varphi \in [0, 2\pi)$

- ▶  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- ▶  $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$

# The Pauli matrices and their eigenstates

name	matrix	eigenstates
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 0\rangle,  1\rangle$

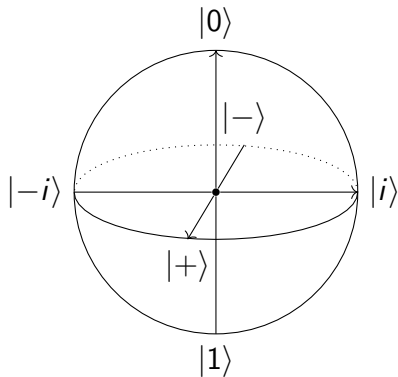
where  $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$  and  
 $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$



# The Pauli matrices and their eigenstates

name	matrix	eigenstates
$X$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$ +\rangle,  -\rangle$
$Z$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 0\rangle,  1\rangle$

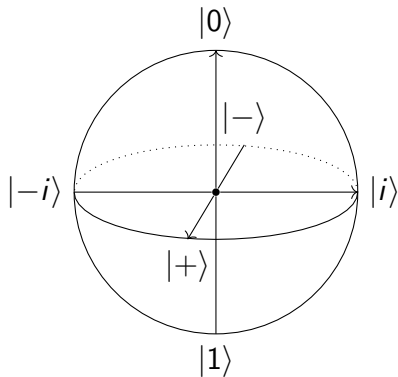
where  $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$  and  
 $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$



# The Pauli matrices and their eigenstates

name	matrix	eigenstates
X	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$ +\rangle,  -\rangle$
Y	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$ i\rangle,  -i\rangle$
Z	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 0\rangle,  1\rangle$

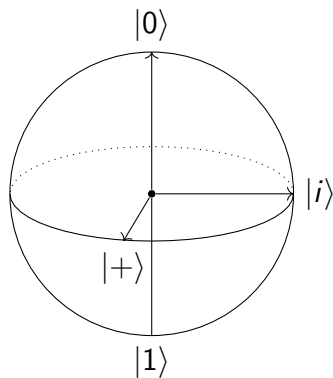
where  $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$  and  
 $|\pm i\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm i|1\rangle)$



# Single-qubit transformations

Quantum transformations are **unitary linear maps**, i.e. maps  $U$  which satisfy  $U^\dagger U = UU^\dagger = I$ , where  $U^\dagger$  is the Hermitian conjugate (or conjugate transpose). Important examples include:

- ▶ The Pauli matrices





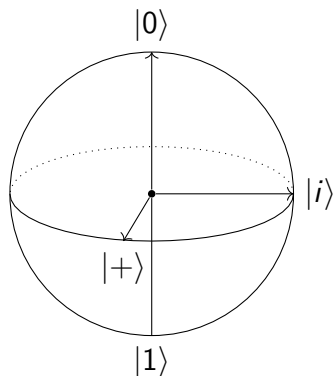
# Single-qubit transformations

Quantum transformations are **unitary linear maps**, i.e. maps  $U$  which satisfy  $U^\dagger U = UU^\dagger = I$ , where  $U^\dagger$  is the Hermitian conjugate (or conjugate transpose).

Important examples include:

- ▶ The Pauli matrices
- ▶ The **phase gates** for  $\xi \in [0, 2\pi)$ :

$$P(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\xi} \end{pmatrix}$$



# Single-qubit transformations

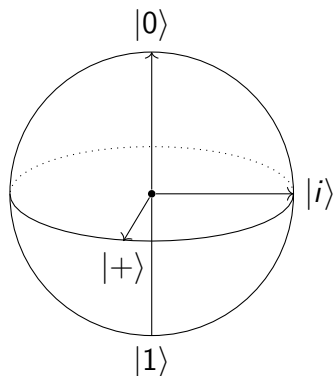
Quantum transformations are **unitary linear maps**, i.e. maps  $U$  which satisfy  $U^\dagger U = UU^\dagger = I$ , where  $U^\dagger$  is the Hermitian conjugate (or conjugate transpose). Important examples include:

- ▶ The Pauli matrices
- ▶ The **phase gates** for  $\xi \in [0, 2\pi)$ :

$$P(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\xi} \end{pmatrix}$$

- ▶ The **Hadamard gate**:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



# Single-qubit transformations

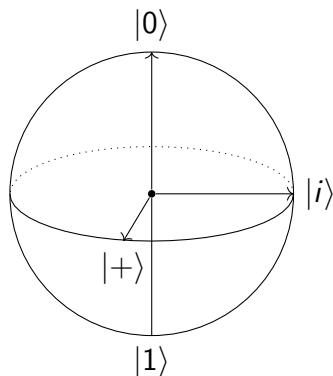
Quantum transformations are **unitary linear maps**, i.e. maps  $U$  which satisfy  $U^\dagger U = U U^\dagger = I$ , where  $U^\dagger$  is the Hermitian conjugate (or conjugate transpose). Important examples include:

- ▶ The Pauli matrices
- ▶ The **phase gates** for  $\xi \in [0, 2\pi)$ :

$$P(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\xi} \end{pmatrix}$$

- ▶ The **Hadamard gate**:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



# Single-qubit transformations

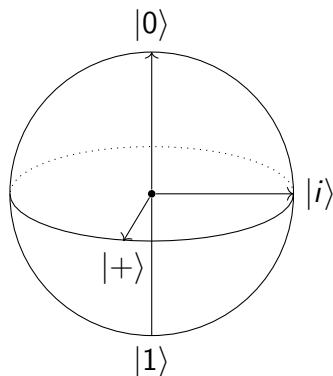
Quantum transformations are **unitary linear maps**, i.e. maps  $U$  which satisfy  $U^\dagger U = UU^\dagger = I$ , where  $U^\dagger$  is the Hermitian conjugate (or conjugate transpose). Important examples include:

- ▶ The Pauli matrices
- ▶ The **phase gates** for  $\xi \in [0, 2\pi)$ :

$$P(\xi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\xi} \end{pmatrix}$$

- ▶ The **Hadamard gate**:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



Phase gates & Hadamard together are universal for single-qubit transformations.

## Two-qubit states

To specify the state of two bits, we write down each bit individually: 00, 01, 10 or 11.

## Two-qubit states

To specify the state of two bits, we write down each bit individually: 00, 01, 10 or 11.

The joint state space of two qubits is the Hilbert space  $\mathbb{C}^2 \otimes \mathbb{C}^2 \simeq \mathbb{C}^4$ ; a state can be written as a superposition over all the 2-bit strings:

$$|\psi\rangle = \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle = \sum_{\mathbf{x} \in \{0,1\}^2} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where normalisation requires  $\sum_{\mathbf{x} \in \{0,1\}^2} |\alpha_{\mathbf{x}}|^2 = 1$ .

## Two-qubit states

To specify the state of two bits, we write down each bit individually: 00, 01, 10 or 11.

The joint state space of two qubits is the Hilbert space  $\mathbb{C}^2 \otimes \mathbb{C}^2 \simeq \mathbb{C}^4$ ; a state can be written as a superposition over all the 2-bit strings:

$$|\psi\rangle = \begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \begin{pmatrix} \alpha_{00} \\ \alpha_{01} \\ \alpha_{10} \\ \alpha_{11} \end{pmatrix} = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle = \sum_{\mathbf{x} \in \{0,1\}^2} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where normalisation requires  $\sum_{\mathbf{x} \in \{0,1\}^2} |\alpha_{\mathbf{x}}|^2 = 1$ .

The states  $\{|\mathbf{x}\rangle\}_{\mathbf{x} \in \{0,1\}^n}$  are called the **computational basis**.

## Product states and entangled states

Some two-qubit states arise as tensor products of single-qubit states, e.g.:

▶  $|0\rangle \otimes |0\rangle = |00\rangle$

▶  $|1\rangle \otimes |-\rangle = |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$

These are called **product states**.



## Product states and entangled states

Some two-qubit states arise as tensor products of single-qubit states, e.g.:

$$\blacktriangleright |0\rangle \otimes |0\rangle = |00\rangle$$

$$\blacktriangleright |1\rangle \otimes |-\rangle = |1\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle)$$

These are called **product states**.

There are also states which cannot be expressed as a tensor product of any pair of single-qubit states, e.g. the 'Bell state':

$$|\Phi_+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Such states are called **entangled**.

## Multi-qubit states

The state of  $n$  qubits lives in the Hilbert space  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$  and is written as a superposition over all the  $n$ -bit strings:

$$|\psi\rangle = \sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where  $\sum_{\mathbf{x} \in \{0,1\}^n} |\alpha_{\mathbf{x}}|^2 = 1$ .

## Multi-qubit states

The state of  $n$  qubits lives in the Hilbert space  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$  and is written as a superposition over all the  $n$ -bit strings:

$$|\psi\rangle = \sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where  $\sum_{\mathbf{x} \in \{0,1\}^n} |\alpha_{\mathbf{x}}|^2 = 1$ .

An  $n$ -qubit state is called:

- ▶ a product state if it can be written as a tensor product of single-qubit states,

## Multi-qubit states

The state of  $n$  qubits lives in the Hilbert space  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$  and is written as a superposition over all the  $n$ -bit strings:

$$|\psi\rangle = \sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where  $\sum_{\mathbf{x} \in \{0,1\}^n} |\alpha_{\mathbf{x}}|^2 = 1$ .

An  $n$ -qubit state is called:

- ▶ a product state if it can be written as a tensor product of single-qubit states,
- ▶ genuinely entangled if it cannot be written as a tensor product at all,

## Multi-qubit states

The state of  $n$  qubits lives in the Hilbert space  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$  and is written as a superposition over all the  $n$ -bit strings:

$$|\psi\rangle = \sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$$

where  $\sum_{\mathbf{x} \in \{0,1\}^n} |\alpha_{\mathbf{x}}|^2 = 1$ .

An  $n$ -qubit state is called:

- ▶ a product state if it can be written as a tensor product of single-qubit states,
- ▶ genuinely entangled if it cannot be written as a tensor product at all,
- ▶ partly entangled if it can be written as a tensor product (but not necessarily of single-qubit states).

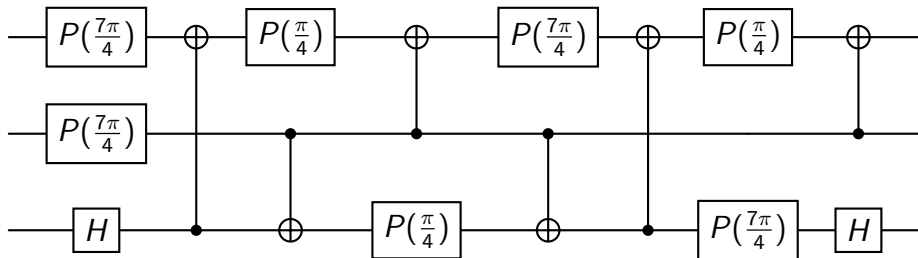
## Multi-qubit transformations and quantum circuits

Reversible classical gates such as NOT, CNOT, and Toffoli are also unitary.

# Multi-qubit transformations and quantum circuits

Reversible classical gates such as NOT, CNOT, and Toffoli are also unitary.

Gates can be composed into quantum circuits:



# Universality, and the fine line to classical simulability

**Theorem** (Barenco et al., 1995)

CNOT, phase gates, and Hadamard together are universal.



# Universality, and the fine line to classical simulability

## Theorem (Barenco et al., 1995)

CNOT, phase gates, and Hadamard together are universal.

## Theorem (Solovay 1995, Kitaev 1997)

CNOT, Hadamard and  $T = P(\frac{\pi}{4}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$  together are universal in the sense that any unitary operation can be efficiently approximated to arbitrary accuracy by a circuit over these gates.

# Universality, and the fine line to classical simulability

## Theorem (Barenco et al., 1995)

CNOT, phase gates, and Hadamard together are universal.

## Theorem (Solovay 1995, Kitaev 1997)

CNOT, Hadamard and  $T = P(\frac{\pi}{4}) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$  together are universal in the sense that any unitary operation can be efficiently approximated to arbitrary accuracy by a circuit over these gates.

## Theorem (Gottesmann & Knill, 1998)

Circuits over CNOT, Hadamard and  $S = P(\frac{\pi}{2}) = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$  are efficiently classically simulable. (They are called Clifford circuits or stabiliser circuits.)

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

## Dirac notation for row vectors and inner product

Given a vector  $|\psi\rangle = (\alpha_0 \ \alpha_1 \ \dots \ \alpha_{k-1})^T$ , its Hermitian conjugate is the 'bra'

$$\langle\psi| = (|\psi\rangle)^\dagger = (\alpha_0^* \ \alpha_1^* \ \dots \ \alpha_{k-1}^*)$$

## Dirac notation for row vectors and inner product

Given a vector  $|\psi\rangle = (\alpha_0 \ \alpha_1 \ \dots \ \alpha_{k-1})^T$ , its Hermitian conjugate is the 'bra'

$$\langle\psi| = (|\psi\rangle)^\dagger = (\alpha_0^* \ \alpha_1^* \ \dots \ \alpha_{k-1}^*)$$

Given a second vector  $|\phi\rangle = (\beta_0 \ \beta_1 \ \dots \ \beta_{k-1})^T$ , the inner product of  $|\psi\rangle$  and  $|\phi\rangle$  is written as the following 'braket':

$$\langle\psi|\phi\rangle = (\alpha_0^* \ \alpha_1^* \ \dots \ \alpha_{k-1}^*) \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{pmatrix} = \sum_{j=0}^{k-1} \alpha_j^* \beta_j$$

## Dirac notation for row vectors and inner product

Given a vector  $|\psi\rangle = (\alpha_0 \ \alpha_1 \ \dots \ \alpha_{k-1})^T$ , its Hermitian conjugate is the 'bra'

$$\langle\psi| = (|\psi\rangle)^\dagger = (\alpha_0^* \ \alpha_1^* \ \dots \ \alpha_{k-1}^*)$$

Given a second vector  $|\phi\rangle = (\beta_0 \ \beta_1 \ \dots \ \beta_{k-1})^T$ , the inner product of  $|\psi\rangle$  and  $|\phi\rangle$  is written as the following 'bracket':

$$\langle\psi|\phi\rangle = (\alpha_0^* \ \alpha_1^* \ \dots \ \alpha_{k-1}^*) \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{k-1} \end{pmatrix} = \sum_{j=0}^{k-1} \alpha_j^* \beta_j$$

The outer product can be written as a 'ketbra'  $|\phi\rangle\langle\psi|$ . With both vectors equal,  $|\psi\rangle\langle\psi|$  is the projector onto the vector space spanned by  $|\psi\rangle$ .

## Observables and quantum measurement

It is impossible to 'read out' the state vector directly. To gain information about a quantum state, need to perform a measurement, which is most commonly described by an **observable**: a Hermitian linear map.

## Observables and quantum measurement

It is impossible to 'read out' the state vector directly. To gain information about a quantum state, need to perform a measurement, which is most commonly described by an **observable**: a Hermitian linear map.

Write such an observable as  $O = \sum_{\lambda} \lambda P_{\lambda}$ , where  $\lambda$  are the eigenvalues and  $P_{\lambda}$  are the projectors onto the corresponding eigenspaces.



# Observables and quantum measurement

It is impossible to 'read out' the state vector directly. To gain information about a quantum state, need to perform a measurement, which is most commonly described by an **observable**: a Hermitian linear map.

Write such an observable as  $O = \sum_{\lambda} \lambda P_{\lambda}$ , where  $\lambda$  are the eigenvalues and  $P_{\lambda}$  are the projectors onto the corresponding eigenspaces.

Measuring the observable  $O$  on state  $|\psi\rangle$  has the following effects:

- ▶ With probability  $p_{\lambda} = \langle \psi | P_{\lambda} | \psi \rangle$ , it produces the outcome  $\lambda$ .

# Observables and quantum measurement

It is impossible to 'read out' the state vector directly. To gain information about a quantum state, need to perform a measurement, which is most commonly described by an **observable**: a Hermitian linear map.

Write such an observable as  $O = \sum_{\lambda} \lambda P_{\lambda}$ , where  $\lambda$  are the eigenvalues and  $P_{\lambda}$  are the projectors onto the corresponding eigenspaces.

Measuring the observable  $O$  on state  $|\psi\rangle$  has the following effects:

- ▶ With probability  $p_{\lambda} = \langle\psi| P_{\lambda} |\psi\rangle$ , it produces the outcome  $\lambda$ .
- ▶ The state of the quantum system is simultaneously projected into the corresponding eigenspace, i.e. post-measurement, the system is in the state

$$\frac{P_{\lambda} |\psi\rangle}{\sqrt{p_{\lambda}}}$$

## Example: $Z$ observable and computational basis measurements

The most common measurement on a single qubit is associated with the observable  $Z = |0\rangle\langle 0| + (-1) |1\rangle\langle 1|$ .

## Example: $Z$ observable and computational basis measurements

The most common measurement on a single qubit is associated with the observable  $Z = |0\rangle\langle 0| + (-1)|1\rangle\langle 1|$ .

Suppose a qubit is in the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Then measuring  $Z$  has the following effect:

- ▶ With probability  $p_{+1} = \langle\psi|0\rangle\langle 0|\psi\rangle = |\alpha|^2$ , the outcome is  $+1$  and the qubit is left in the state  $|0\rangle$ .

## Example: $Z$ observable and computational basis measurements

The most common measurement on a single qubit is associated with the observable  $Z = |0\rangle\langle 0| + (-1)|1\rangle\langle 1|$ .

Suppose a qubit is in the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Then measuring  $Z$  has the following effect:

- ▶ With probability  $p_{+1} = \langle\psi|0\rangle\langle 0|\psi\rangle = |\alpha|^2$ , the outcome is  $+1$  and the qubit is left in the state  $|0\rangle$ .
- ▶ With probability  $p_{-1} = \langle\psi|1\rangle\langle 1|\psi\rangle = |\beta|^2$ , the outcome is  $-1$  and the qubit is left in the state  $|1\rangle$ .

## Example: $Z$ observable and computational basis measurements

The most common measurement on a single qubit is associated with the observable  $Z = |0\rangle\langle 0| + (-1)|1\rangle\langle 1|$ .

Suppose a qubit is in the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Then measuring  $Z$  has the following effect:

- ▶ With probability  $p_{+1} = \langle\psi|0\rangle\langle 0|\psi\rangle = |\alpha|^2$ , the outcome is  $+1$  and the qubit is left in the state  $|0\rangle$ .
- ▶ With probability  $p_{-1} = \langle\psi|1\rangle\langle 1|\psi\rangle = |\beta|^2$ , the outcome is  $-1$  and the qubit is left in the state  $|1\rangle$ .

## Example: $Z$ observable and computational basis measurements

The most common measurement on a single qubit is associated with the observable  $Z = |0\rangle\langle 0| + (-1)|1\rangle\langle 1|$ .

Suppose a qubit is in the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Then measuring  $Z$  has the following effect:

- ▶ With probability  $p_{+1} = \langle\psi|0\rangle\langle 0|\psi\rangle = |\alpha|^2$ , the outcome is  $+1$  and the qubit is left in the state  $|0\rangle$ .
- ▶ With probability  $p_{-1} = \langle\psi|1\rangle\langle 1|\psi\rangle = |\beta|^2$ , the outcome is  $-1$  and the qubit is left in the state  $|1\rangle$ .

Instead of the labels  $\pm 1$ , we often use labels  $0, 1$  and call this a 'computational basis measurement': i.e. we write  $p_0 = |\alpha|^2$  and  $p_1 = |\beta|^2$ .

## Example: $X$ -measurement

The Pauli- $X$  matrix is also an observable  $X = |+\rangle\langle+| + (-1) |-\rangle\langle-|$ . Suppose a qubit is in the state  $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$ . Measuring  $X$  has the following effect:

- ▶ With probability

$$p_{+1} = \langle\psi|+\rangle \langle+|\psi\rangle = |\langle+|\psi\rangle|^2 = \left| \frac{1}{\sqrt{2}}(\alpha + \beta) \right|^2$$

the outcome is  $+1$  and the qubit is left in the state  $|+\rangle$ .



## Example: $X$ -measurement

The Pauli- $X$  matrix is also an observable  $X = |+\rangle\langle+| + (-1)|-\rangle\langle-|$ . Suppose a qubit is in the state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ . Measuring  $X$  has the following effect:

- ▶ With probability

$$p_{+1} = \langle\psi|+\rangle\langle+|\psi\rangle = |\langle+|\psi\rangle|^2 = \left|\frac{1}{\sqrt{2}}(\alpha + \beta)\right|^2$$

the outcome is  $+1$  and the qubit is left in the state  $|+\rangle$ .

- ▶ With probability

$$p_{-1} = |\langle-|\psi\rangle|^2 = \left|\frac{1}{\sqrt{2}}(\alpha - \beta)\right|^2$$

the outcome is  $-1$  and the qubit is left in the state  $|-\rangle$ .

## Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes.

## Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

## Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ .

# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle$$

# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle = (|\psi'\rangle)^{\dagger} P_{\lambda} |\psi'\rangle$$

# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle = (|\psi'\rangle)^{\dagger} P_{\lambda} |\psi'\rangle = (\gamma |\psi\rangle)^{\dagger} P_{\lambda} (\gamma |\psi\rangle)$$

# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle = (|\psi'\rangle)^{\dagger} P_{\lambda} |\psi'\rangle = (\gamma |\psi\rangle)^{\dagger} P_{\lambda} (\gamma |\psi\rangle) = \gamma^* \gamma \langle \psi | P_{\lambda} | \psi \rangle$$



# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle = (|\psi'\rangle)^{\dagger} P_{\lambda} |\psi'\rangle = (\gamma |\psi\rangle)^{\dagger} P_{\lambda} (\gamma |\psi\rangle) = \gamma^* \gamma \langle \psi | P_{\lambda} | \psi \rangle = \langle \psi | P_{\lambda} | \psi \rangle$$

# Some implications of quantum measurements

Most information stored in a quantum state cannot be retrieved

The state of a qubit is described by two complex numbers (or at least two real numbers) but measurement gives 2 discrete outcomes. An  $n$ -qubit state is described by  $2^n$  complex numbers but the (up to)  $2^n$  possible measurement outcomes can be described using at most  $n$  bits.

Global factors are physically irrelevant

Suppose  $|\psi'\rangle = \gamma |\psi\rangle$ , where  $\gamma \in \mathbb{C}$  satisfies  $|\gamma| = 1$ . Then for any observable  $O = \sum_{\lambda} \lambda P_{\lambda}$ , the probability of outcome  $\lambda$  satisfies

$$\langle \psi' | P_{\lambda} | \psi' \rangle = (|\psi'\rangle)^{\dagger} P_{\lambda} |\psi'\rangle = (\gamma |\psi\rangle)^{\dagger} P_{\lambda} (\gamma |\psi\rangle) = \gamma^* \gamma \langle \psi | P_{\lambda} | \psi \rangle = \langle \psi | P_{\lambda} | \psi \rangle$$

The same holds for any more general form of measurement, justifying the assumption we made when introducing the Bloch sphere picture.

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

## A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

**A selection of quantum algorithms**

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

## Quantum implementations of Boolean functions

Given a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ , we can build a quantum circuit on  $(n + m)$  qubits implementing the unitary linear map

$$U_f(|\mathbf{x}\rangle |\mathbf{y}\rangle) = |\mathbf{x}\rangle |\mathbf{y} + f(\mathbf{x})\rangle$$

where  $\mathbf{x} \in \{0, 1\}^n$ ,  $\mathbf{y} \in \{0, 1\}^m$  and the sum  $\mathbf{y} + f(\mathbf{x})$  interprets the bit strings as binary numbers and is taken modulo  $2^m$ .

## Quantum implementations of Boolean functions

Given a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ , we can build a quantum circuit on  $(n + m)$  qubits implementing the unitary linear map

$$U_f(|\mathbf{x}\rangle |\mathbf{y}\rangle) = |\mathbf{x}\rangle |\mathbf{y} + f(\mathbf{x})\rangle$$

where  $\mathbf{x} \in \{0, 1\}^n$ ,  $\mathbf{y} \in \{0, 1\}^m$  and the sum  $\mathbf{y} + f(\mathbf{x})$  interprets the bit strings as binary numbers and is taken modulo  $2^m$ . The first  $n$  qubits are called the **input register** and the last  $m$  qubits are called the **output register**.

## Quantum implementations of Boolean functions

Given a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ , we can build a quantum circuit on  $(n + m)$  qubits implementing the unitary linear map

$$U_f(|\mathbf{x}\rangle |\mathbf{y}\rangle) = |\mathbf{x}\rangle |\mathbf{y} + f(\mathbf{x})\rangle$$

where  $\mathbf{x} \in \{0, 1\}^n$ ,  $\mathbf{y} \in \{0, 1\}^m$  and the sum  $\mathbf{y} + f(\mathbf{x})$  interprets the bit strings as binary numbers and is taken modulo  $2^m$ . The first  $n$  qubits are called the **input register** and the last  $m$  qubits are called the **output register**.

E.g. if  $f$  is AND, then  $U_f$  is the Toffoli gate, which for any  $x_1, x_2, y \in \{0, 1\}$  acts as

$$U_{\text{AND}}(|x_1 x_2\rangle |y\rangle) = |x_1 x_2\rangle |y \oplus (x_1 \wedge x_2)\rangle$$

## Deutsch's problem

**Input:** A 'black box' implementation of a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , which is promised to be either constant or balanced.

**Output:** Decide with certainty whether  $f$  is constant or balanced.

'Black box' means the only way to interact with the implementation is to enter an input and read out the corresponding output: this is called a **query**.



## Deutsch's problem

**Input:** A 'black box' implementation of a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , which is promised to be either constant or balanced.

**Output:** Decide with certainty whether  $f$  is constant or balanced.

'Black box' means the only way to interact with the implementation is to enter an input and read out the corresponding output: this is called a **query**.

Classically, in the worst case, need  $2^{n-1} + 1$  queries for certainty.

# Deutsch's problem

**Input:** A 'black box' implementation of a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , which is promised to be either constant or balanced.

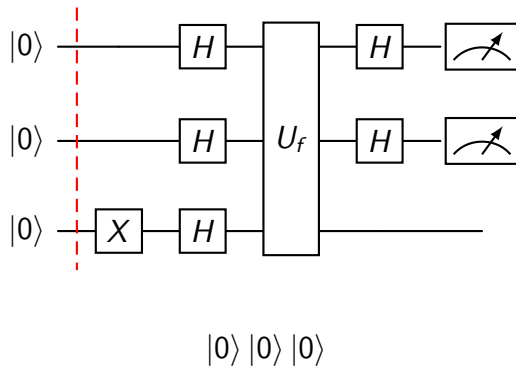
**Output:** Decide with certainty whether  $f$  is constant or balanced.

'Black box' means the only way to interact with the implementation is to enter an input and read out the corresponding output: this is called a **query**.

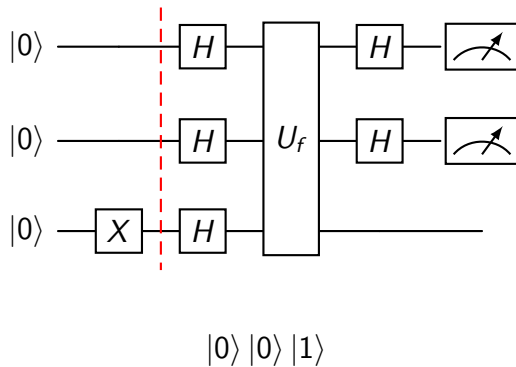
Classically, in the worst case, need  $2^{n-1} + 1$  queries for certainty.

If the implementation is quantum, the Deutsch-Jozsa algorithm shows a single query is enough.

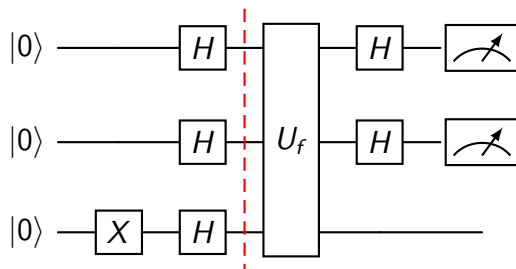
# The Deutsch-Jozsa algorithm



# The Deutsch-Jozsa algorithm



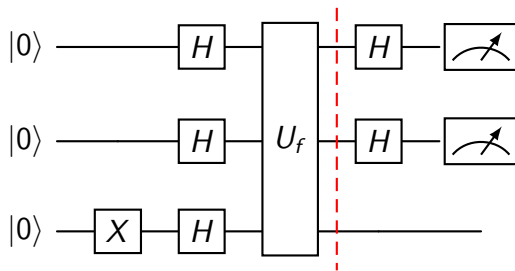
# The Deutsch-Jozsa algorithm



$$|+\rangle |+\rangle |-\rangle = \frac{1}{2\sqrt{2}} \sum_{\mathbf{x} \in \{0,1\}^2} |\mathbf{x}\rangle (|0\rangle - |1\rangle)$$

For any  $n$ , we have  $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle$ .

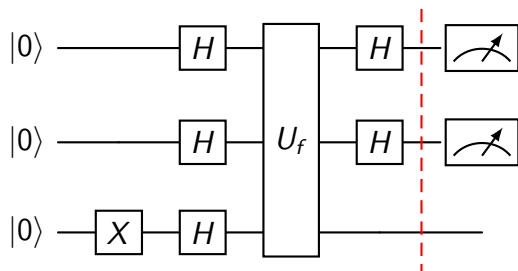
# The Deutsch-Jozsa algorithm



$$\frac{1}{2\sqrt{2}} \sum_{\mathbf{x} \in \{0,1\}^2} |\mathbf{x}\rangle (|0 \oplus f(\mathbf{x})\rangle - |1 \oplus f(\mathbf{x})\rangle) = \frac{1}{2} \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle$$

This way of moving the value of a function to the exponent of a scalar is called **phase kickback**.

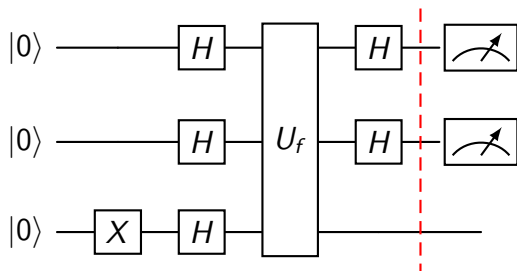
# The Deutsch-Jozsa algorithm



$$(H^{\otimes 2} \otimes I) \frac{1}{2} \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle = \frac{1}{4} \sum_{\mathbf{x} \in \{0,1\}^2} \sum_{\mathbf{y} \in \{0,1\}^2} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{y}} |\mathbf{y}\rangle |-\rangle$$

Note:  $H|\mathbf{x}\rangle = \frac{1}{\sqrt{2}} \sum_{\mathbf{y} \in \{0,1\}} (-1)^{\mathbf{x} \cdot \mathbf{y}} |\mathbf{y}\rangle$  for any  $\mathbf{x} \in \{0,1\}$ .

# The Deutsch-Jozsa algorithm

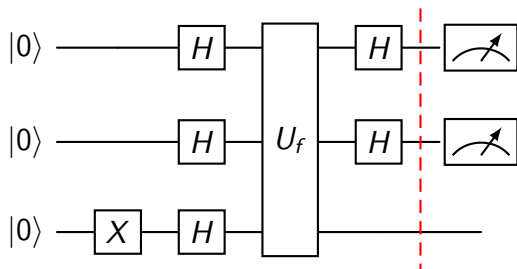


$$(H^{\otimes 2} \otimes I) \frac{1}{2} \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle = \frac{1}{4} \sum_{\mathbf{y} \in \{0,1\}^2} \left( \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{y}} \right) |\mathbf{y}\rangle |-\rangle$$

If  $f$  is constant,  $\left| \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{0}} \right| = 4$ ; if  $f$  is balanced, this sum is 0.



# The Deutsch-Jozsa algorithm



$$(H^{\otimes 2} \otimes I) \frac{1}{2} \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle |-\rangle = \frac{1}{4} \sum_{\mathbf{y} \in \{0,1\}^2} \left( \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{y}} \right) |\mathbf{y}\rangle |-\rangle$$

If  $f$  is constant,  $\left| \sum_{\mathbf{x} \in \{0,1\}^2} (-1)^{f(\mathbf{x}) + \mathbf{x} \cdot \mathbf{0}} \right| = 4$ ; if  $f$  is balanced, this sum is 0. This means  $p_{00} = 1$  if  $f$  is constant,  $p_{00} = 0$  if  $f$  is balanced; so one query suffices.

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

## A selection of quantum algorithms

The Deutsch-Jozsa algorithm

Quantum Fourier transform and Shor's algorithm

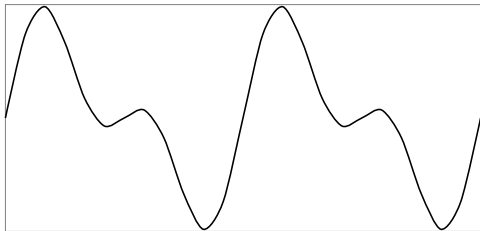
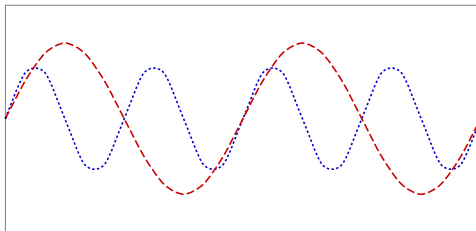
Grover's algorithm

Quantum Teleportation

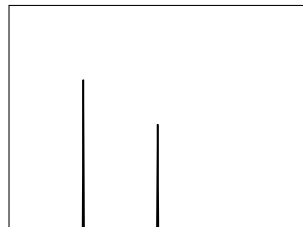
Optimisation of quantum computations using the ZX-calculus

Conclusions

# Fourier transform: intuition



time domain



frequency domain

# The discrete Fourier transform

Given a complex vector  $(x_0, \dots, x_{N-1})$  of fixed length  $N$ , its discrete Fourier transform is the vector  $(y_0, \dots, y_{N-1})$  defined for any  $0 \leq k < N$  as

$$y_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j.$$

## The discrete Fourier transform

Given a complex vector  $(x_0, \dots, x_{N-1})$  of fixed length  $N$ , its discrete Fourier transform is the vector  $(y_0, \dots, y_{N-1})$  defined for any  $0 \leq k < N$  as

$$y_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} x_j.$$

The classical Fast Fourier Transform algorithm (FFT) runs in  $O(N \log N)$ .

# The discrete Fourier transform

Given a complex vector  $(x_0, \dots, x_{N-1})$  of fixed length  $N$ , its discrete Fourier transform is the vector  $(y_0, \dots, y_{N-1})$  defined for any  $0 \leq k < N$  as

$$y_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k / N} x_j.$$

The classical Fast Fourier Transform algorithm (FFT) runs in  $O(N \log N)$ .

The quantum Fourier transform is a discrete Fourier transform on the amplitudes of the state vector:

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \quad \text{or equivalently} \quad \sum_{j=0}^{N-1} x_j |j\rangle \mapsto \sum_{k=0}^{N-1} y_k |k\rangle$$

# The discrete Fourier transform

Given a complex vector  $(x_0, \dots, x_{N-1})$  of fixed length  $N$ , its discrete Fourier transform is the vector  $(y_0, \dots, y_{N-1})$  defined for any  $0 \leq k < N$  as

$$y_k := \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k / N} x_j.$$

The classical Fast Fourier Transform algorithm (FFT) runs in  $O(N \log N)$ .

The quantum Fourier transform is a discrete Fourier transform on the amplitudes of the state vector:

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle \quad \text{or equivalently} \quad \sum_{j=0}^{N-1} x_j |j\rangle \mapsto \sum_{k=0}^{N-1} y_k |k\rangle$$

If  $N = 2^n$ , this uses  $n$  qubits.

# The quantum Fourier transform

Suppose  $N = 2^n$  and write  $j$  in binary:  $j_1j_2 \dots j_n \in \{0, 1\}^n$  corresponding to the number  $\sum_{\ell=1}^n j_\ell 2^{n-\ell}$ .



# The quantum Fourier transform

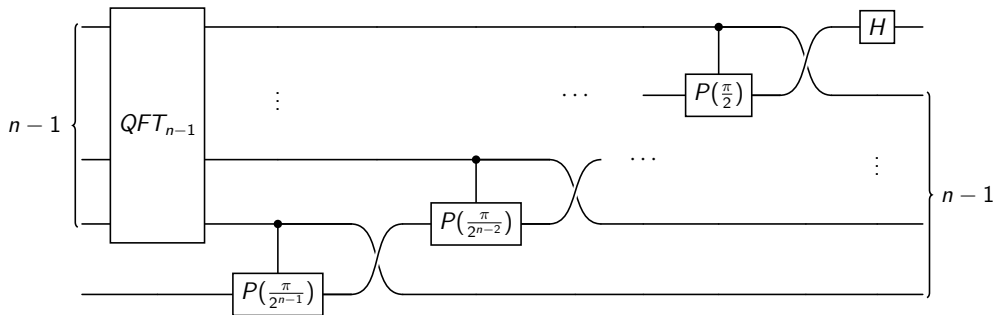
Suppose  $N = 2^n$  and write  $j$  in binary:  $j_1 j_2 \dots j_n \in \{0, 1\}^n$  corresponding to the number  $\sum_{\ell=1}^n j_\ell 2^{n-\ell}$ . Then we can write  $|j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle$  as

$$|j_1 \dots j_n\rangle \mapsto \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i j_n / 2} |1\rangle) (|0\rangle + e^{2\pi i (j_{n-1}/2 + j_n/4)} |1\rangle) \dots (|0\rangle + e^{2\pi i \sum_{\ell=1}^n j_\ell 2^{-\ell}} |1\rangle)$$

# The quantum Fourier transform

Suppose  $N = 2^n$  and write  $j$  in binary:  $j_1 j_2 \dots j_n \in \{0, 1\}^n$  corresponding to the number  $\sum_{\ell=1}^n j_\ell 2^{n-\ell}$ . Then we can write  $|j\rangle \mapsto \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i j k / 2^n} |k\rangle$  as

$$|j_1 \dots j_n\rangle \mapsto \frac{1}{\sqrt{2^n}} (|0\rangle + e^{2\pi i j_n / 2} |1\rangle) (|0\rangle + e^{2\pi i (j_{n-1}/2 + j_n/4)} |1\rangle) \dots (|0\rangle + e^{2\pi i \sum_{\ell=1}^n j_\ell 2^{-\ell}} |1\rangle)$$



# The quantum period finding problem

Suppose  $1 \leq r < \sqrt{2^n}$ . An  $n$ -qubit state 'has period  $r$ ' if it is of the form

$$|\psi_{r,x_0}\rangle := \frac{1}{\sqrt{A}} \sum_{\ell=0}^{A-1} |x_0 + \ell r\rangle$$

where  $x_0$  is a random offset in the range  $0 \leq x_0 < r$  that may be different for each state produced, and  $A$  is the smallest integer such that  $x_0 + Ar \geq 2^n$ .

# The quantum period finding problem

Suppose  $1 \leq r < \sqrt{2^n}$ . An  $n$ -qubit state 'has period  $r$ ' if it is of the form

$$|\psi_{r,x_0}\rangle := \frac{1}{\sqrt{A}} \sum_{\ell=0}^{A-1} |x_0 + \ell r\rangle$$

where  $x_0$  is a random offset in the range  $0 \leq x_0 < r$  that may be different for each state produced, and  $A$  is the smallest integer such that  $x_0 + Ar \geq 2^n$ .

**Input:** a black box producing quantum states  $|\psi_{r,x_0}\rangle$  for some unknown fixed  $r$ , and a method for checking whether a guess for  $r$  is correct

**Output:** the period  $r$

# The quantum period finding problem

Suppose  $1 \leq r < \sqrt{2^n}$ . An  $n$ -qubit state 'has period  $r$ ' if it is of the form

$$|\psi_{r,x_0}\rangle := \frac{1}{\sqrt{A}} \sum_{\ell=0}^{A-1} |x_0 + \ell r\rangle$$

where  $x_0$  is a random offset in the range  $0 \leq x_0 < r$  that may be different for each state produced, and  $A$  is the smallest integer such that  $x_0 + Ar \geq 2^n$ .

**Input:** a black box producing quantum states  $|\psi_{r,x_0}\rangle$  for some unknown fixed  $r$ , and a method for checking whether a guess for  $r$  is correct

**Output:** the period  $r$

This can be solved using the QFT.

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),
- ▶  $N$  is odd (otherwise 2 is a non-trivial factor and the problem is solved), and



# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),
- ▶  $N$  is odd (otherwise 2 is a non-trivial factor and the problem is solved), and
- ▶  $N$  cannot be written as  $N = a^b$  for any integers  $a \geq 1$ ,  $b \geq 2$  (this check runs in polynomial time and, if applicable, outputs the non-trivial factor  $a$ ).

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),
- ▶  $N$  is odd (otherwise 2 is a non-trivial factor and the problem is solved), and
- ▶  $N$  cannot be written as  $N = a^b$  for any integers  $a \geq 1$ ,  $b \geq 2$  (this check runs in polynomial time and, if applicable, outputs the non-trivial factor  $a$ ).

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),
- ▶  $N$  is odd (otherwise 2 is a non-trivial factor and the problem is solved), and
- ▶  $N$  cannot be written as  $N = a^b$  for any integers  $a \geq 1$ ,  $b \geq 2$  (this check runs in polynomial time and, if applicable, outputs the non-trivial factor  $a$ ).

Then solve the **Order Finding problem**:

**Input:** a number  $x$  in the range  $1 < x < N - 1$  such that  $\gcd(x, N) = 1$

**Output:** the smallest positive  $r$  such that  $x^r \equiv 1 \pmod{N}$

# Shor's algorithm for the Factoring problem

**Input:** a positive integer  $N$ , which is promised to be a composite number

**Output:** an integer  $p$  in the range  $1 < p \leq \sqrt{N}$  such that  $p$  divides  $N$

First verify:

- ▶  $N$  is not prime (this can be done in polynomial time),
- ▶  $N$  is odd (otherwise 2 is a non-trivial factor and the problem is solved), and
- ▶  $N$  cannot be written as  $N = a^b$  for any integers  $a \geq 1$ ,  $b \geq 2$  (this check runs in polynomial time and, if applicable, outputs the non-trivial factor  $a$ ).

Then solve the **Order Finding problem**:

**Input:** a number  $x$  in the range  $1 < x < N - 1$  such that  $\gcd(x, N) = 1$

**Output:** the smallest positive  $r$  such that  $x^r \equiv 1 \pmod{N}$

It is likely that  $r$  is even and one of  $\gcd(x^{r/2} \pm 1, N)$  is a non-trivial factor.

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

## A selection of quantum algorithms

The Deutsch-Jozsa algorithm

Quantum Fourier transform and Shor's algorithm

**Grover's algorithm**

Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Grover's search problem

**Input:** a (quantum) black box implementing some Boolean function

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

**Output:** a bit string  $\mathbf{x} \in \{0, 1\}^n$  such that  $f(\mathbf{x}) = 1$

# Grover's search problem

**Input:** a (quantum) black box implementing some Boolean function

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

**Output:** a bit string  $\mathbf{x} \in \{0, 1\}^n$  such that  $f(\mathbf{x}) = 1$

Let  $A = \{\mathbf{x} \in \{0, 1\}^n \mid f(\mathbf{x}) = 1\}$  and set  $M = |A|$ ,  $N = 2^n$ .

## Grover's search problem

**Input:** a (quantum) black box implementing some Boolean function  
 $f : \{0, 1\}^n \rightarrow \{0, 1\}$

**Output:** a bit string  $\mathbf{x} \in \{0, 1\}^n$  such that  $f(\mathbf{x}) = 1$

Let  $A = \{\mathbf{x} \in \{0, 1\}^n \mid f(\mathbf{x}) = 1\}$  and set  $M = |A|$ ,  $N = 2^n$ .

Classically, need  $O(N/M)$  queries on average to find an element of  $A$ .



## Grover's search problem

**Input:** a (quantum) black box implementing some Boolean function  
 $f : \{0, 1\}^n \rightarrow \{0, 1\}$

**Output:** a bit string  $\mathbf{x} \in \{0, 1\}^n$  such that  $f(\mathbf{x}) = 1$

Let  $A = \{\mathbf{x} \in \{0, 1\}^n \mid f(\mathbf{x}) = 1\}$  and set  $M = |A|$ ,  $N = 2^n$ .

Classically, need  $O(N/M)$  queries on average to find an element of  $A$ .

Quantumly,  $O(\sqrt{N/M})$  queries suffice if  $M$  is known and  $M \ll N$ .

# Grover's search problem

**Input:** a (quantum) black box implementing some Boolean function  
 $f : \{0, 1\}^n \rightarrow \{0, 1\}$

**Output:** a bit string  $\mathbf{x} \in \{0, 1\}^n$  such that  $f(\mathbf{x}) = 1$

Let  $A = \{\mathbf{x} \in \{0, 1\}^n \mid f(\mathbf{x}) = 1\}$  and set  $M = |A|$ ,  $N = 2^n$ .

Classically, need  $O(N/M)$  queries on average to find an element of  $A$ .

Quantumly,  $O(\sqrt{N/M})$  queries suffice if  $M$  is known and  $M \ll N$ .

Combination of Grover's algorithm and QFT can also be used to determine  $M$  if it is unknown: this is 'quantum counting'.

## Phase kickback and a useful subspace

The quantum black box is given as  $U_f(|\mathbf{x}\rangle |y\rangle) = |\mathbf{x}\rangle |y \oplus f(x)\rangle$ , but we can use the 'phase kickback trick' from Deutsch-Jozsa algorithm to turn it into  $U'_f(|\mathbf{x}\rangle |-\rangle) = (-1)^{f(x)} |\mathbf{x}\rangle |-\rangle$ .

## Phase kickback and a useful subspace

The quantum black box is given as  $U_f(|\mathbf{x}\rangle |y\rangle) = |\mathbf{x}\rangle |y \oplus f(x)\rangle$ , but we can use the 'phase kickback trick' from Deutsch-Jozsa algorithm to turn it into  $U'_f(|\mathbf{x}\rangle |-\rangle) = (-1)^{f(x)} |\mathbf{x}\rangle |-\rangle$ .

Consider the 2-dimensional vector space spanned by

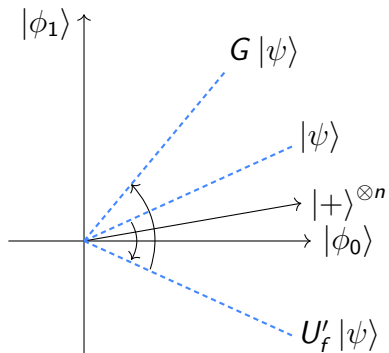
$$|\phi_1\rangle = \frac{1}{\sqrt{M}} \sum_{\mathbf{x} \in A} |\mathbf{x}\rangle \quad \text{and} \quad |\phi_0\rangle = \frac{1}{\sqrt{N-M}} \sum_{\mathbf{x} \in \{0,1\}^n \setminus A} |\mathbf{x}\rangle$$

This space also contains

$$|+\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{\mathbf{x} \in \{0,1\}^n} |\mathbf{x}\rangle = \sqrt{\frac{N-M}{N}} |\phi_0\rangle + \sqrt{\frac{M}{N}} |\phi_1\rangle$$

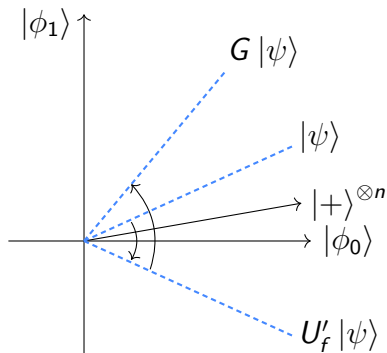
# The Grover operator

Let  $G = U_+ U'_f$ , where  $U_+ = 2|++\dots+\rangle\langle ++\dots+| - I$ .



# The Grover operator

Let  $G = U_+ U'_f$ , where  $U_+ = 2|++\dots+\rangle\langle ++\dots+| - I$ .



$G$  performs a rotation by angle  $\theta \approx 2\sqrt{M/N}$ , so after  $O(\sqrt{N/M})$  applications, probability of measuring a state in  $A$  is high. Checking correctness is easy.

# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

## A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation**

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Transmitting quantum information without a quantum channel

Suppose Alice wants to send a quantum state to Bob, but she can only send bits, not qubits.

- ▶ If Alice knows the state, she can send a classical description: but this would require a lot of data and still be approximate.



# Transmitting quantum information without a quantum channel

Suppose Alice wants to send a quantum state to Bob, but she can only send bits, not qubits.

- ▶ If Alice knows the state, she can send a classical description: but this would require a lot of data and still be approximate.
- ▶ Yet with some advance preparation, Alice can send even an unknown quantum state using only 2 bits of communication.

# Transmitting quantum information without a quantum channel

Suppose Alice wants to send a quantum state to Bob, but she can only send bits, not qubits.

- ▶ If Alice knows the state, she can send a classical description: but this would require a lot of data and still be approximate.
- ▶ Yet with some advance preparation, Alice can send even an unknown quantum state using only 2 bits of communication.

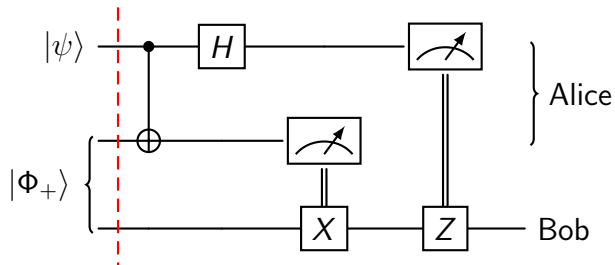
# Transmitting quantum information without a quantum channel

Suppose Alice wants to send a quantum state to Bob, but she can only send bits, not qubits.

- ▶ If Alice knows the state, she can send a classical description: but this would require a lot of data and still be approximate.
- ▶ Yet with some advance preparation, Alice can send even an unknown quantum state using only 2 bits of communication.

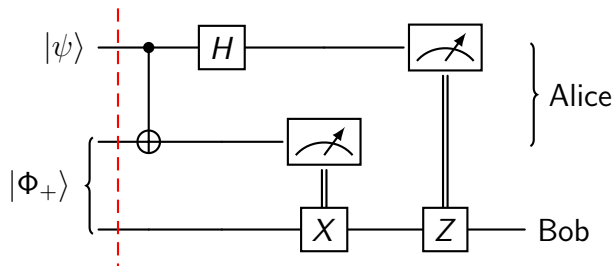
Alice and Bob need arrange ahead of time to share an entangled Bell state  $|\Phi_+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ .

# The quantum teleportation protocol



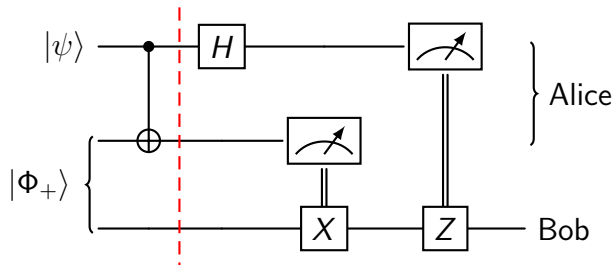
$$\begin{aligned}
 & (H \otimes I_2)(\text{CNOT} \otimes I) |\psi\rangle |\Phi_+\rangle \\
 &= (H \otimes I_2)(\text{CNOT} \otimes I) \frac{1}{\sqrt{2}} (\alpha |0\rangle + \beta |1\rangle) (|00\rangle + |11\rangle)
 \end{aligned}$$

# The quantum teleportation protocol



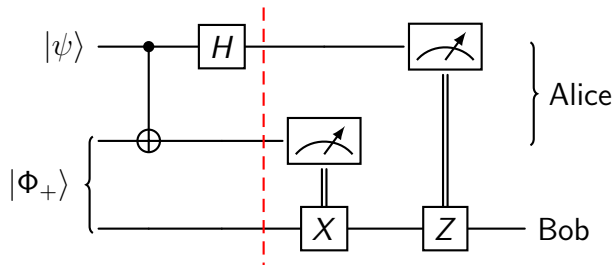
$$\begin{aligned}
 & (H \otimes I_2)(\text{CNOT} \otimes I) |\psi\rangle |\Phi_+\rangle \\
 &= (H \otimes I_2)(\text{CNOT} \otimes I) \frac{1}{\sqrt{2}} (\alpha |000\rangle + \alpha |011\rangle + \beta |100\rangle + \beta |111\rangle)
 \end{aligned}$$

# The quantum teleportation protocol



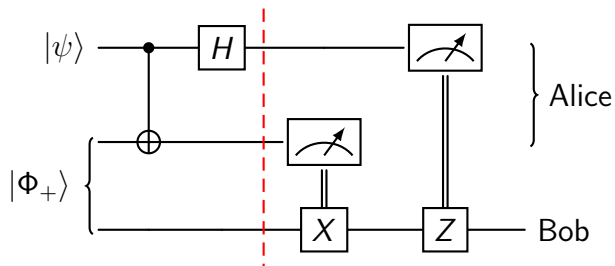
$$\begin{aligned} & (H \otimes I_2)(\text{CNOT} \otimes I) |\psi\rangle |\Phi_+\rangle \\ &= (H \otimes I_2) \frac{1}{\sqrt{2}} (\alpha |000\rangle + \alpha |011\rangle + \beta |110\rangle + \beta |101\rangle) \end{aligned}$$

# The quantum teleportation protocol



$$\begin{aligned}
 & (H \otimes I_2)(\text{CNOT} \otimes I) |\psi\rangle |\Phi_+\rangle \\
 &= \frac{1}{2}(\alpha |000\rangle + \alpha |100\rangle + \alpha |011\rangle + \alpha |111\rangle + \beta |010\rangle - \beta |110\rangle + \beta |001\rangle - \beta |101\rangle)
 \end{aligned}$$

# The quantum teleportation protocol



$$\begin{aligned}
 & (H \otimes I_2)(\text{CNOT} \otimes I) |\psi\rangle |\Phi_+\rangle \\
 &= \frac{1}{2} \left( |00\rangle (\alpha |0\rangle + \beta |1\rangle) + |01\rangle (\alpha |1\rangle + \beta |0\rangle) + |10\rangle (\alpha |0\rangle - \beta |1\rangle) \right. \\
 & \qquad \qquad \qquad \left. + |11\rangle (\alpha |1\rangle - \beta |0\rangle) \right)
 \end{aligned}$$



# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

## Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

## Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?

## Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?
- ▶ How can we check that two given circuits describe the same linear map?

## Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?
- ▶ How can we check that two given circuits describe the same linear map?

# Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?
- ▶ How can we check that two given circuits describe the same linear map?

For example, we might want

- ▶ a circuit with fewer gates in total, or

# Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?
- ▶ How can we check that two given circuits describe the same linear map?

For example, we might want

- ▶ a circuit with fewer gates in total, or
- ▶ a circuit with fewer layers of gates, or

# Motivation: optimisation & equality checking

Quantum computational resources are limited, so we need to use them efficiently.

- ▶ Given a quantum circuit, can we find a more efficient circuit that describes the same linear map?
- ▶ How can we check that two given circuits describe the same linear map?

For example, we might want

- ▶ a circuit with fewer gates in total, or
- ▶ a circuit with fewer layers of gates, or
- ▶ a circuit with fewer of a specific type of gate.

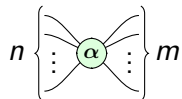


# The ZX-calculus components: (mostly) spiders instead of gates

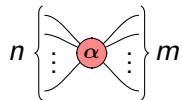
Hadamard gate



Z-spider



X-spider

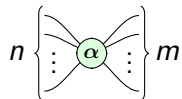


# The ZX-calculus components: (mostly) spiders instead of gates

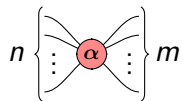
Hadamard gate

$$\text{---} \square \text{---} \rightsquigarrow |+\rangle\langle 0| + |-\rangle\langle 1| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Z-spider



X-spider



# The ZX-calculus components: (mostly) spiders instead of gates

Hadamard gate

$$\text{---} \square \text{---} \rightsquigarrow |+\rangle\langle 0| + |-\rangle\langle 1| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Z-spider

$$n \left\{ \begin{array}{c} \diagup \quad \diagdown \\ \vdots \\ \alpha \\ \vdots \\ \diagdown \quad \diagup \end{array} \right\} m \rightsquigarrow \underbrace{|0\dots 0\rangle}_m \underbrace{\langle 0\dots 0|}_n + e^{i\alpha} \underbrace{|1\dots 1\rangle}_m \underbrace{\langle 1\dots 1|}_n = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix}$$

X-spider

$$n \left\{ \begin{array}{c} \diagup \quad \diagdown \\ \vdots \\ \alpha \\ \vdots \\ \diagdown \quad \diagup \end{array} \right\} m$$

# The ZX-calculus components: (mostly) spiders instead of gates

Hadamard gate

$$\text{---} \square \text{---} \rightsquigarrow |+\rangle\langle 0| + |-\rangle\langle 1| = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Z-spider

$$n \left\{ \begin{array}{c} \diagup \quad \diagdown \\ \vdots \quad \vdots \\ \alpha \\ \vdots \quad \vdots \\ \diagdown \quad \diagup \end{array} \right\} m \rightsquigarrow \underbrace{|0\dots 0\rangle}_m \underbrace{\langle 0\dots 0|}_n + e^{i\alpha} \underbrace{|1\dots 1\rangle}_m \underbrace{\langle 1\dots 1|}_n = \begin{pmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e^{i\alpha} \end{pmatrix}$$

X-spider

$$n \left\{ \begin{array}{c} \diagup \quad \diagdown \\ \vdots \quad \vdots \\ \alpha \\ \vdots \quad \vdots \\ \diagdown \quad \diagup \end{array} \right\} m \rightsquigarrow \underbrace{|+\dots +\rangle}_m \underbrace{\langle +\dots +|}_n + e^{i\alpha} \underbrace{|-\dots -\rangle}_m \underbrace{\langle -\dots -|}_n$$

## Wires in the ZX-calculus

$$\text{————} \quad \rightsquigarrow \quad |0\rangle\langle 0| + |1\rangle\langle 1| \quad = \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

## Wires in the ZX-calculus

$$\begin{array}{l} \text{---} \\ \text{X} \end{array} \quad \rightsquigarrow \quad \begin{array}{l} |0\rangle\langle 0| + |1\rangle\langle 1| \\ |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11| \end{array} \quad = \quad \begin{array}{l} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

## Wires in the ZX-calculus

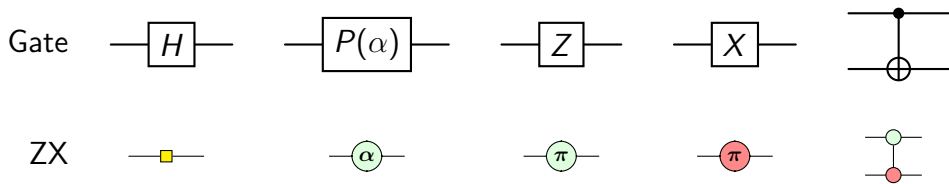
$$\begin{array}{l} \text{---} \\ \text{X} \\ \text{)} \end{array} \quad \rightsquigarrow \quad \begin{array}{l} |0\rangle\langle 0| + |1\rangle\langle 1| \\ |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11| \\ \langle 00| + \langle 11| \end{array} \quad = \quad \begin{array}{l} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ (1 \ 0 \ 0 \ 1) \end{array}$$

## Wires in the ZX-calculus

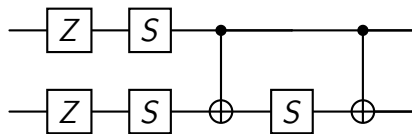
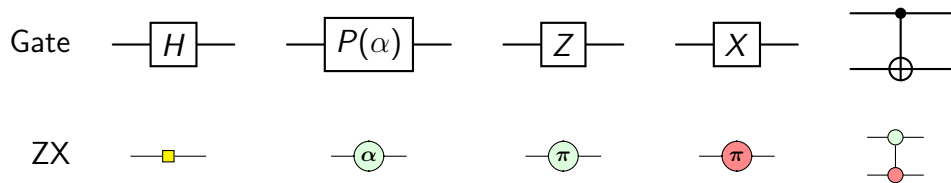
$$\begin{array}{l} \text{---} \quad \rightsquigarrow \quad |0\rangle\langle 0| + |1\rangle\langle 1| \quad = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \\ \text{X} \quad \rightsquigarrow \quad |00\rangle\langle 00| + |10\rangle\langle 01| + |01\rangle\langle 10| + |11\rangle\langle 11| \quad = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ \\ \text{)} \quad \rightsquigarrow \quad \langle 00| + \langle 11| \quad = (1 \ 0 \ 0 \ 1) \\ \\ \text{(} \quad \rightsquigarrow \quad |00\rangle + |11\rangle \quad = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{array}$$



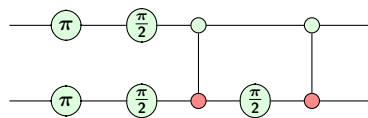
# Translating circuits into ZX-diagrams



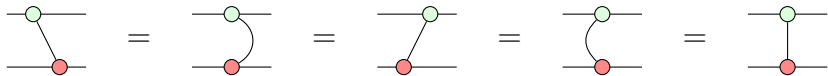
# Translating circuits into ZX-diagrams



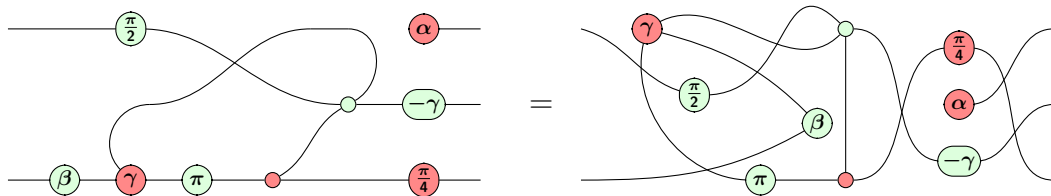
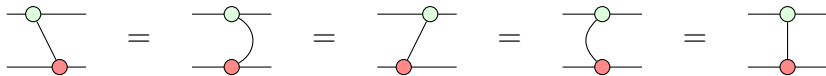
$\rightsquigarrow$



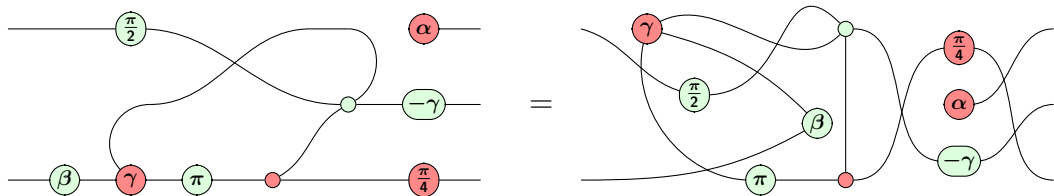
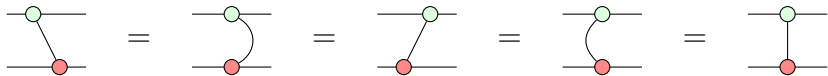
## Only connectivity matters



# Only connectivity matters

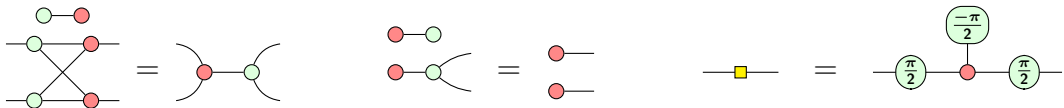
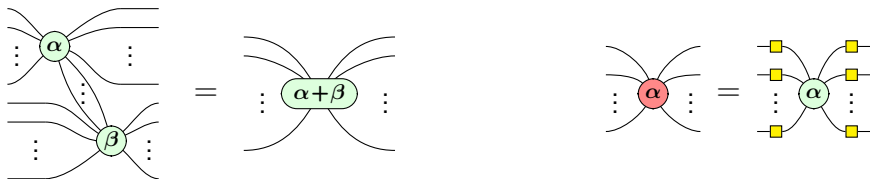


# Only connectivity matters

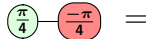


This is made mathematically rigorous using monoidal category theory.

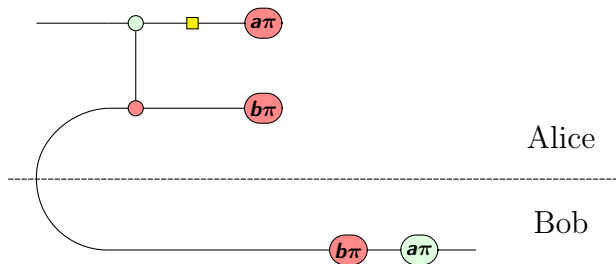
# A complete set of ZX-calculus rewrite rules



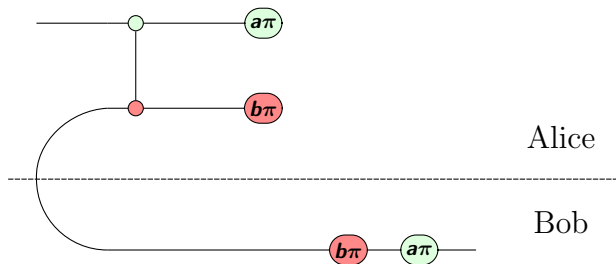
Only connectivity matters.



# Example: quantum teleportation

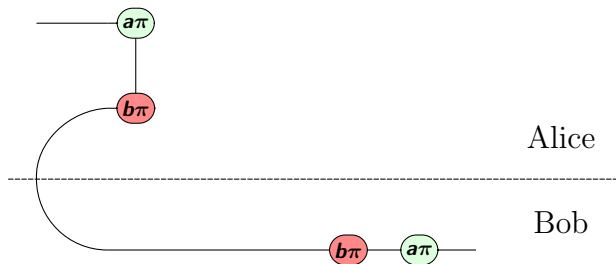


## Example: quantum teleportation

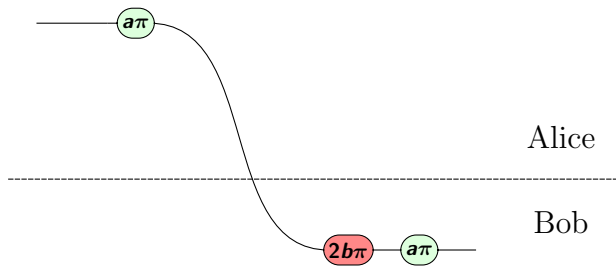




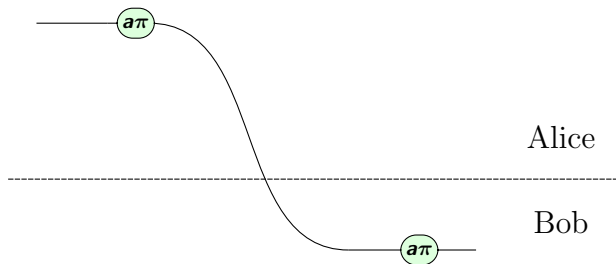
## Example: quantum teleportation



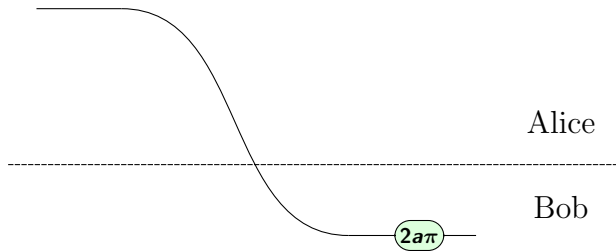
## Example: quantum teleportation



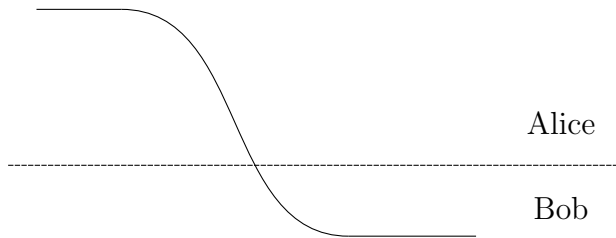
## Example: quantum teleportation



## Example: quantum teleportation



## Example: quantum teleportation



# Outline

Some non-quantum computer science

Quantum computing basics: states and transformations

Quantum computing basics: measurements

A selection of quantum algorithms

- The Deutsch-Jozsa algorithm

- Quantum Fourier transform and Shor's algorithm

- Grover's algorithm

- Quantum Teleportation

Optimisation of quantum computations using the ZX-calculus

Conclusions

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$

## Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$



# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols
- ▶ optimisation of quantum computations using ZX-calculus is an area of active research

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols
- ▶ optimisation of quantum computations using ZX-calculus is an area of active research

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols
- ▶ optimisation of quantum computations using ZX-calculus is an area of active research

Topics not discussed:

- ▶ quantum error correction

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols
- ▶ optimisation of quantum computations using ZX-calculus is an area of active research

Topics not discussed:

- ▶ quantum error correction
- ▶ quantum simulation

# Summary

- ▶ qubit states are vectors  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$  in  $\mathbb{C}^2$
- ▶ states of multiple qubits are vectors  $\sum_{\mathbf{x} \in \{0,1\}^n} \alpha_{\mathbf{x}} |\mathbf{x}\rangle$  in  $(\mathbb{C}^2)^{\otimes n} \simeq \mathbb{C}^{2^n}$
- ▶ qubit transformations are unitary linear maps, they can be expressed as circuits using CNOT and single-qubit gates
- ▶ quantum measurements are probabilistic and change the state
- ▶ there are useful quantum algorithms and protocols
- ▶ optimisation of quantum computations using ZX-calculus is an area of active research

Topics not discussed:

- ▶ quantum error correction
- ▶ quantum simulation
- ▶ building physical quantum computers