Quantum Computation (CO484)

Quantum Gates and Circuits

Herbert Wiklicky

herbert@doc.ic.ac.uk

Autumn 2017

Classical Gates

At heart of classical (electronic) circuits we have to consider gates like for example:

Classical Gates

At heart of classical (electronic) circuits we have to consider gates like for example:

AND $\equiv \wedge$		
0	0	0
0	1	0
1	0	0
1	1	1

Classical Gates

At heart of classical (electronic) circuits we have to consider gates like for example:

AND $\equiv \wedge$		
0	0	0
0	1	0
1	0	0
1	1	1

XOR $\equiv \oplus$		
0	0	0
0	1	1
1	0	1
1	1	0

Classical Gates

At heart of classical (electronic) circuits we have to consider gates like for example:

AND $\equiv \wedge$		
0	0	0
0	1	0
1	0	0
1	1	1

XOR $\equiv \oplus$		
0	0	0
0	1	1
1	0	1
1	1	0

NAND		
0	0	1
0	1	1
1	0	1
1	1	0

Classical Gates

At heart of classical (electronic) circuits we have to consider gates like for example:

AND $\equiv \wedge$		
0	0	0
0	1	0
1	0	0
1	1	1

XOR $\equiv \oplus$		
0	0	0
0	1	1
1	0	1
1	1	0

NAND		
0	0	1
0	1	1
1	0	1
1	1	0

The idea is to define similar quantum gates, taking two (or n) qubits at input and producing some output. Contrary to classical gates we have to use **unitary**, i.e. reversible, gates in quantum circuits.

The Controlled-NOT or CNOT Gate

The quantum analog of a classical XOR-gate is the CNOT-gate.

The Controlled-NOT or CNOT Gate

The quantum analog of a classical XOR-gate is the CNOT-gate. The behaviour of the CNOT-gate (on two qubits, i.e. $\mathbb{C}^2 \otimes \mathbb{C}^2$), is for base vectors $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$:

$$|x, y\rangle \mapsto |x, y \oplus x\rangle \quad \text{with} \quad y \oplus x = (y + x) \bmod 2$$

i.e. $|00\rangle \mapsto |00\rangle, |01\rangle \mapsto |01\rangle, |10\rangle \mapsto |11\rangle, |11\rangle \mapsto |10\rangle$.

The Controlled-NOT or CNOT Gate

The quantum analog of a classical XOR-gate is the CNOT-gate. The behaviour of the CNOT-gate (on two qubits, i.e. $\mathbb{C}^2 \otimes \mathbb{C}^2$), is for base vectors $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$:

$$|x, y\rangle \mapsto |x, y \oplus x\rangle \quad \text{with} \quad y \oplus x = (y + x) \bmod 2$$

i.e. $|00\rangle \mapsto |00\rangle, |01\rangle \mapsto |01\rangle, |10\rangle \mapsto |11\rangle, |11\rangle \mapsto |10\rangle$.

We represent the CNOT-gate graphically and as a matrix (with respect to the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$) as:

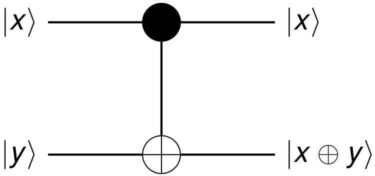
The Controlled-NOT or CNOT Gate

The quantum analog of a classical XOR-gate is the CNOT-gate.
The behaviour of the CNOT-gate (on two qubits, i.e. $\mathbb{C}^2 \otimes \mathbb{C}^2$), is for base vectors $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$:

$$|x, y\rangle \mapsto |x, y \oplus x\rangle \quad \text{with} \quad y \oplus x = (y + x) \bmod 2$$

i.e. $|00\rangle \mapsto |00\rangle, |01\rangle \mapsto |01\rangle, |10\rangle \mapsto |11\rangle, |11\rangle \mapsto |10\rangle$.

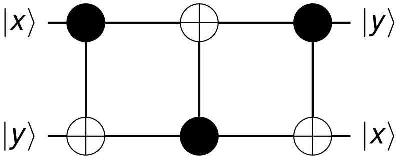
We represent the CNOT-gate graphically and as a matrix (with respect to the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$) as:



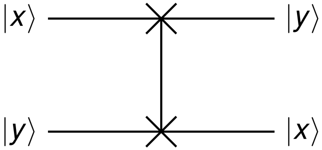
$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Swapping Gate

We can exploit the CNOT-Gate to SWAP two qubits:



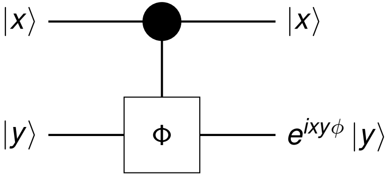
is depicted by (shorthand):



Exercise: Check that this really maps $|x\rangle \otimes |y\rangle$ into $|y\rangle \otimes |x\rangle$ (for all $|x\rangle$ and $|y\rangle$ not just base vectors?).

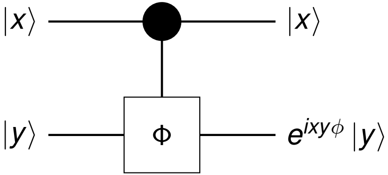
Controlled Phase Gate

The controlled phase-gate is depicted as follows (for base vectors $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$):



Controlled Phase Gate

The controlled phase-gate is depicted as follows (for base vectors $|x\rangle, |y\rangle \in \{|0\rangle, |1\rangle\}$):



Its matrix/operator representation is given by:

$$\Phi = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\phi} \end{pmatrix}$$

on any two qubits, i.e. vectors in $\mathbb{C}^2 \otimes \mathbb{C}^2$.

General Controlled Gate

In general, we can control any single qubit transformation $\mathbf{U} : \mathbb{C}^2 \to \mathbb{C}^2$ by another qubit, i.e. such that for all $|y\rangle \in \mathbb{C}^2$:

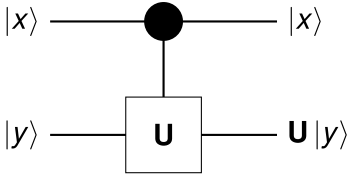
$$\begin{aligned} |0\rangle \otimes |y\rangle &\mapsto |0\rangle \otimes |y\rangle \\ |1\rangle \otimes |y\rangle &\mapsto |1\rangle \otimes \mathbf{U}\,|y\rangle \end{aligned}$$

General Controlled Gate

In general, we can control any single qubit transformation $\mathbf{U} : \mathbb{C}^2 \to \mathbb{C}^2$ by another qubit, i.e. such that for all $|y\rangle \in \mathbb{C}^2$:

$$\begin{array}{rcl} |0\rangle \otimes |y\rangle & \mapsto & |0\rangle \otimes |y\rangle \\ |1\rangle \otimes |y\rangle & \mapsto & |1\rangle \otimes \mathbf{U}\,|y\rangle \end{array}$$

The diagrammatic representation is:



Toffoli Gate

The Toffoli-gate is a 3-qubit quantum gate on $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 = \mathbb{C}^8$ with the following behaviour $\mathbf{T} : |x, y, z\rangle \mapsto |x', y', z'\rangle$ and matrix representation (standard base enumeration):

input			*output*		
x	y	z	x'	y'	z'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

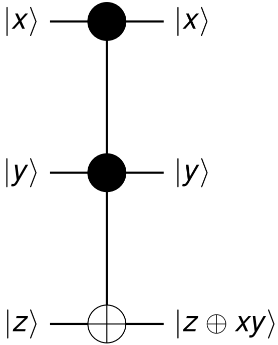
$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Toffoli Gate Usage

The Toffoli gate can be used can be used to implement a reversible version of NAND and a FANOUT gate.

Toffoli Gate Usage

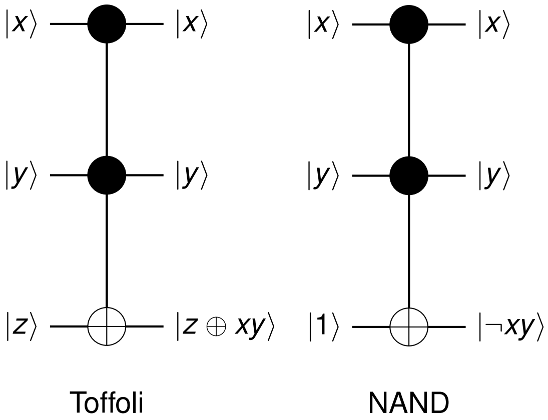
The Toffoli gate can be used can be used to implement a reversible version of NAND and a FANOUT gate.



Toffoli

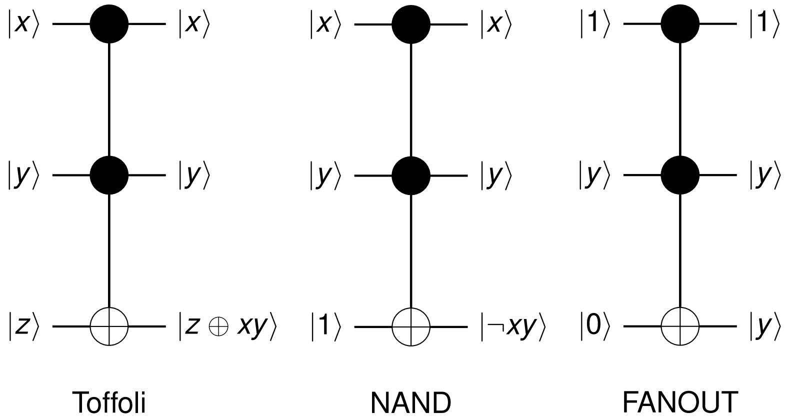
Toffoli Gate Usage

The Toffoli gate can be used can be used to implement a reversible version of NAND and a FANOUT gate.



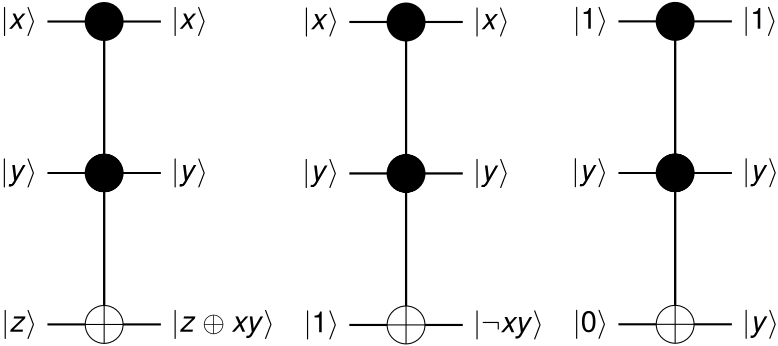
Toffoli Gate Usage

The Toffoli gate can be used can be used to implement a reversible version of NAND and a FANOUT gate.



Toffoli Gate Usage

The Toffoli gate can be used can be used to implement a reversible version of NAND and a FANOUT gate.



Toffoli NAND FANOUT

This works only with $x, y \in \{0, 1\}$.

Linear Maps from Functions

In general, we can take any (binary) function

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

and define a corresponding linear map $\mathbf{T}_f$

$$\mathbf{T}_f : (\mathcal{V}(\{0,1\}))^{\otimes n} \to (\mathcal{V}(\{0,1\}))^{\otimes m} \text{ or } \mathbf{T}_f : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$$

Linear Maps from Functions

In general, we can take any (binary) function

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

and define a corresponding linear map $\mathbf{T}_f$

$$\mathbf{T}_f : (\mathcal{V}(\{0, 1\}))^{\otimes n} \to (\mathcal{V}(\{0, 1\}))^{\otimes m} \text{ or } \mathbf{T}_f : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$$

We just have to read the map f as an instruction on how **base** vectors should be transformed under $\mathbf{T}_f$ (into base vectors).

Linear Maps from Functions

In general, we can take any (binary) function

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

and define a corresponding linear map $\mathbf{T}_f$

$$\mathbf{T}_f : (\mathcal{V}(\{0,1\}))^{\otimes n} \to (\mathcal{V}(\{0,1\}))^{\otimes m} \text{ or } \mathbf{T}_f : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$$

We just have to read the map f as an instruction on how **base** vectors should be transformed under $\mathbf{T}_f$ (into base vectors).

Once we know or specify the image of all base vectors we know the (matrix representation) of $\mathbf{T}_f$ via

$$\mathbf{T}_f \, |x\rangle = |f(x)\rangle$$

E.g. with $f(011) = 10101$ we have $\mathbf{T}_f : |011\rangle \mapsto |10101\rangle$.

Linear Maps from Functions

In general, we can take any (binary) function

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

and define a corresponding linear map $\mathbf{T}_f$

$$\mathbf{T}_f : (\mathcal{V}(\{0,1\}))^{\otimes n} \to (\mathcal{V}(\{0,1\}))^{\otimes m} \text{ or } \mathbf{T}_f : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes m}$$

We just have to read the map f as an instruction on how **base** vectors should be transformed under $\mathbf{T}_f$ (into base vectors).

Once we know or specify the image of all base vectors we know the (matrix representation) of $\mathbf{T}_f$ via

$$\mathbf{T}_f \left|x\right\rangle = \left|f(x)\right\rangle$$

E.g. with $f(011) = 10101$ we have $\mathbf{T}_f : \left|011\right\rangle \mapsto \left|10101\right\rangle$.

Problem: $\mathbf{T}_f$ is, in general, not unitary, i.e. reversible.

Reversible Operators from General Functions

Reversibility makes it impossible to have a quantum device $\mathbf{U}_f$ which **just** computes a general function f, i.e. $\mathbf{U}_f : |x\rangle \mapsto |f(x)\rangle$.

Reversible Operators from General Functions

Reversibility makes it impossible to have a quantum device $\mathbf{U}_f$ which **just** computes a general function f, i.e. $\mathbf{U}_f : |x\rangle \mapsto |f(x)\rangle$.

However, we can always "pack" up a function f as a unitary operator $\mathbf{U}_f$ using an ancilla qubit to remember the initial state, e.g. $|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f(x)\rangle$.

Reversible Operators from General Functions

Reversibility makes it impossible to have a quantum device $\mathbf{U}_f$ which **just** computes a general function f, i.e. $\mathbf{U}_f : |x\rangle \mapsto |f(x)\rangle$.

However, we can always "pack" up a function f as a unitary operator $\mathbf{U}_f$ using an ancilla qubit to remember the initial state, e.g. $|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f(x)\rangle$. The **standard** implementation of $f : \{0,1\}^n \to \{0,1\}^m$ as unitary operator $\mathbf{U}_f$ on $\mathbb{C}^{2^n} \otimes \mathbb{C}^{2^m}$ is:

$$\mathbf{U}_f : |x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

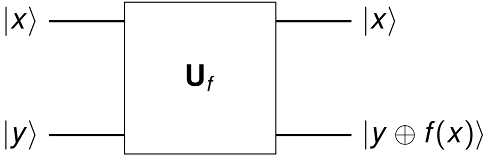
Reversible Operators from General Functions

Reversibility makes it impossible to have a quantum device $\mathbf{U}_f$ which **just** computes a general function f, i.e. $\mathbf{U}_f : |x\rangle \mapsto |f(x)\rangle$.

However, we can always "pack" up a function f as a unitary operator $\mathbf{U}_f$ using an ancilla qubit to remember the initial state, e.g. $|x\rangle \otimes |0\rangle \mapsto |x\rangle \otimes |f(x)\rangle$. The **standard** implementation of $f : \{0,1\}^n \to \{0,1\}^m$ as unitary operator $\mathbf{U}_f$ on $\mathbb{C}^{2^n} \otimes \mathbb{C}^{2^m}$ is:

$$\mathbf{U}_f : |x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |y \oplus f(x)\rangle$$

Graphically represented by the diagram/quantum circuit:



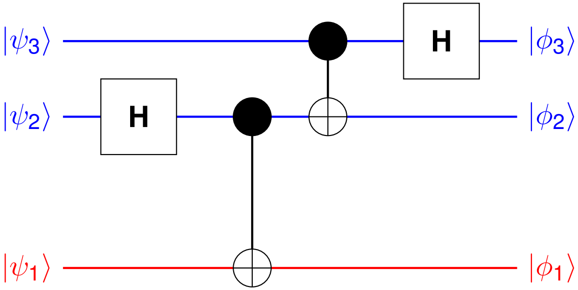
Quantum Circuit Model

We can specify a quantum algorithm on qubit registers – i.e. a unitary operator $\mathbf{U} : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ – using a combination of (standardised) quantum gates – like Hadamard, Pauli, etc. – and maybe "oracles" like $\mathbf{U}_f$ as well as measurements.

Quantum Circuit Model

We can specify a quantum algorithm on qubit registers – i.e. a unitary operator $\mathbf{U} : (\mathbb{C}^2)^{\otimes n} \to (\mathbb{C}^2)^{\otimes n}$ – using a combination of (standardised) quantum gates – like Hadamard, Pauli, etc. – and maybe “oracles” like $\mathbf{U}_f$ as well as measurements.

For example, the quantum circuit for **teleportation** (without correction) as an operator on $(\mathbb{C}^2)^{\otimes 3}$ is given as follows:



Calculations for Small Quantum Circuits

Circuits with few qubits can “implemented”, e.g. in `octave`, etc.

```
q0 = [1,0]'
q1 = [0,1]'

H = (1/sqrt(2))*[1, 1;1,-1]
CX = [1, 0, 0, 0; 0, 1, 0, 0;
      0, 0, 0, 1; 0, 0, 1, 0]

S1 = kron(eye(2),H,eye(2))
S2 = kron(eye(2),CX)
S3 = kron(CX,eye(2))
S4 = kron(H,eye(2),eye(2))

T = S1*S2*S3*S4
```

Computational Expressivness

The question arises: What we can compute with a given set of basic quantum gates? What can we compute with a quantum circuit?

Computational Expressivness

The question arises: What we can compute with a given set of basic quantum gates? What can we compute with a quantum circuit?

For **permutations** it is well known that all permutations can be decomposed into elementary so-called **transpositions** which only exchange two elements.

Computational Expressivness

The question arises: What we can compute with a given set of basic quantum gates? What can we compute with a quantum circuit?

For **permutations** it is well known that all permutations can be decomposed into elementary so-called **transpositions** which only exchange two elements. Similar results also exist for **rotations**.

Computational Expressivness

The question arises: What we can compute with a given set of basic quantum gates? What can we compute with a quantum circuit?

For **permutations** it is well known that all permutations can be decomposed into elementary so-called **transpositions** which only exchange two elements. Similar results also exist for **rotations**.

For general unitary operators **U** on $\mathbb{C}^n$ – in particular on m qubits, i.e. $\mathbb{C}^{2^m} = (\mathbb{C}^2)^{\otimes m}$ – an analogue results gurantees that 2×2 unitary matrices make up all unitary operators.

See e.g.: A. Yu. Kitaev, A. H. Shen, M. N. Vyalyi: Classical and Quantum Computation, AMS, 2002, p70.

Unitary Operators on $\mathbb{C}^n$

Theorem

An arbitrary unitary operator $\mathbf{U}$ *on the space* $\mathbb{C}^n$ *can be represented as a product of* $\frac{n(n-1)}{2}$ *matrices of the form:*

$$\begin{pmatrix} 1 & \dots & 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & \ddots & \vdots & & & & & \vdots \\ 0 & \dots & 1 & & & & & \vdots \\ \vdots & & & a & b & & & \vdots \\ \vdots & & & c & d & & & \vdots \\ \vdots & & & & & 1 & \dots & 0 \\ \vdots & & & & & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 & \dots & 1 \end{pmatrix}$$

with $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *a* 2×2 *unitary matrix (on* $\mathbb{C}^2$*).*

Approximation of Unitary Operators

If we are only interested in “about the right result” we have:

Approximation of Unitary Operators

If we are only interested in "about the right result" we have:

Given two unitary transformations **U** and **V**. The error of approximation is defined by

$$e(\mathbf{U}, \mathbf{V}) = \sup_{|\phi\rangle} \|(\mathbf{U} - \mathbf{V}) |\phi\rangle \|$$

Approximation of Unitary Operators

If we are only interested in “about the right result” we have:

Given two unitary transformations **U** and **V**. The error of approximation is defined by

$$e(\mathbf{U}, \mathbf{V}) = \sup_{|\phi\rangle} \|(\mathbf{U} - \mathbf{V})\, |\phi\rangle\, \|$$

Definition

A set of gates $\mathcal{G} = \{\mathbf{G}_1, \ldots\}$ is said to be approximatly universal if any n-qubit operator **U** (with $n \geq 1$) can be approximated to arbitrary accuracy, i.e. for all $\varepsilon > 0$ there exists a circuit **V** which is constructed of gates in $\mathcal{G}$ and their controlled versions such that we have $e(\mathbf{U}, \mathbf{V}) < \varepsilon$.

(Approximatly) Universal Gates

A possible set of approximatly universal gates (e.g. Kaye, Laflamme, Mosca: Introduction to Quantum Computing, p71):

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \Phi\left(\frac{\pi}{4}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

(Approximatly) Universal Gates

A possible set of approximatly universal gates (e.g. Kaye, Laflamme, Mosca: Introduction to Quantum Computing, p71):

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \Phi\left(\frac{\pi}{4}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Theorem

The set $\mathcal{G} = \{\mathbf{H}, \Phi(\frac{\pi}{4})\}$ is universal for 1-qubits.

(Approximatly) Universal Gates

A possible set of approximatly universal gates (e.g. Kaye, Laflamme, Mosca: Introduction to Quantum Computing, p71):

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad \Phi\left(\frac{\pi}{4}\right) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

$$\mathbf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Theorem

The set $\mathcal{G} = \{\mathbf{H}, \Phi(\frac{\pi}{4})\}$ is universal for 1-qubits.

Theorem

The set $\mathcal{G} = \{\mathbf{CNOT}, \mathbf{H}, \Phi(\frac{\pi}{4})\}$ is a universal set of gates.