SAPIENZA
UNIVERSITÀ DI ROMA

"Sapienza" University of Rome
Faculty of Information Engineering,
Informatics and Statistics
Department of Computer Science

Quantum Computing

Author
Alessio Bandiera

October 21, 2025

Contents

Information and Contacts

Personal notes and summaries collected as part of the *Quantum Computing* course offered by the degree in Computer Science of the University of Rome "La Sapienza".

Further information and notes can be found at the following link: **https://github.com/aflaag-notes**. Anyone can feel free to report inaccuracies, improvements or requests through the Issue system provided by GitHub itself or by contacting the author privately:

- Email: **alessio.bandiera02@gmail.com**
- LinkedIn: **Alessio Bandiera**

The notes are constantly being updated, so please check if the changes have already been made in the most recent version.

Suggested prerequisites:

TODO

Licence:

1

Introduction on Quantum Computation

1.1 The Qubit

Quantum computing is a rapidly developing discipline that explores how the laws of quantum mechanics can be used to *process information*. While classical computation is based on *bits* that take values of either 0 or 1, quantum computation relies on quantum bits, or **qubits**. A qubit can exist in a "superposition" of classical states, allowing it to encode richer information than a single bit. Furthermore, qubits can exhibit particular properties that enable forms of information processing with no classical counterpart. Such properties provide the foundation for algorithms that promise to solve certain problems more efficiently than their classical analogues.

The design of quantum algorithms requires a different perspective from that of classical computation. In classical computer science, the majority of widely studied algorithms are *deterministic*, meaning that for a given input they will always produce the *same output*. Some algorithms are *randomized*, making use of probability to achieve efficiency or simplicity, yet even in those cases the computation itself is ultimately classical in nature. In fact, to achieve such *randomness* classical algorithms employ **pseudo-random number generation**, which must ultimately produce finite sequences.

Quantum computation, by contrast, *incorporates probability* at its core. The act of measuring a quantum system does not reveal a single, predetermined result, but rather yields one outcome from a distribution of possible outcomes, with probabilities governed by the system's quantum state. This fundamental probabilistic characteristic distinguishes quantum algorithms from their classical counterparts.

In fact, in the context of quantum computing we are often interested in **probabilistic algorithms**: for such algorithms, a given input i can lead to a finite set of possible outputs $o_1, \ldots, o_N$, each occurring with an associated probability $p_1, \ldots, p_N$ — where $\sum_{i=1}^{n} p_i = 1$.

As previously mentioned, the quantum equivalent of the classical bits are the **qubit**, but define the qubits we first need to define some preliminary concepts. The following vectors are called **basis states**

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

and they represent the classical bits 0 and 1 respectively — the notation above is called "braket" notation and it will be explored in greater detail in later sections.

So what is a qubit? A qubit is the basic unit of information in quantum computing, which represents a **superposition** of states simultaneously — note that we will refer to qubits and their states interchangeably, since the only thing that we care about a qubit is its own state

In practice, the state of a qubit is a vector

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \alpha\begin{pmatrix}1\\0\end{pmatrix} + \beta\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\alpha\\\beta\end{pmatrix}$$

where $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$ are called **probability amplitudes**. But why are we talking about probabilities in the first place? The "true" state of a qubit **cannot be observed**, and we say that the qubit is in a *superpotion* of $|0\rangle$ and $|1\rangle$ in the sense that α and β describe the probabilities of getting either states once the qubit is measured. This is because to know the value of a qubit we have to *measure it*, and the measurement operation itself will make the qubit *collapse* into either $|0\rangle$ or $|1\rangle$ with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively, i.e.

$$\Pr[\text{measured qubit is } |0\rangle] = |\alpha|^2 \qquad \Pr[\text{measured qubit is } |1\rangle] = |\beta|^2$$

To use a more compact notation, we will denote this property as follows:

$$\alpha\,|0\rangle + \beta\,|1\rangle \begin{cases} |0\rangle & @ \ |\alpha|^2 \\ |1\rangle & @ \ |\beta|^2 \end{cases}$$

where the @ notation (read as "at") denotes the probabilty of the corresponding outcome. Note that if we measure a collapsed qubit we will keep observing the same state indefinitely.

In reality, to be precise qubits actually collapse into any multiple $z\,|0\rangle$ or $z\,|1\rangle$, where $z \in \mathbb{C}$ is a complex number such that $|z| = 1$, but this is not relevant from a physical point of view. In fact, for any θ physicists treat $|\psi\rangle = |0\rangle$ and $|\psi'\rangle = e^{i\theta}\,|0\rangle$ as the *same physical state*, because probabilities depend on squared magnitudes and thus

$$\left|e^{i\theta}\alpha\right|^2 = |\alpha|^2$$

(and the same applies for β too) even though $|\psi\rangle$ and $|\psi'\rangle$ are different vectors mathematically. Therefore, in general we can actually drop the **global phases** from the qubits entirely.

1.2 Qubit operations

What can we do with qubits other then *measure them*? The operations that can be applied on qubits are restricted to **unitary transformations**, which are linear maps that preserve the norms — we will discuss the precise definition in the next chapter. For instance, the identity matrix I is an example of trivial unitary transformation, but also the NOT matrix, which is the following

$$\text{NOT} := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

which has the effect of *swapping* the input basis state

$$\text{NOT}\,|0\rangle = |1\rangle \qquad \text{NOT}\,|1\rangle = |0\rangle$$

This matrix behaves as the classical NOT gate with the usual bits in classical computing, in fact will refer to *transformations* and *gates* interchangeably.

More in general, the NOT operation belongs to a family of operation represented by the so called **Pauli matrices**.

Definition 1.1: Pauli matrices

The **Pauli matrices** are the following four 2×2 matrices:

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \sigma_x := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

In particular, we observe that the second matrix σ_x is exactly the matrix of the NOT operator. We will see the Z and Y operators — representing the other two matrices, respectively — as well in later sections.

Another very important transformation is represented by the **Hadamard gate**, which is the following matrix

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

This matrix has the effect of "mapping" classical states into superpositions:

$$H\,|0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \begin{cases} |0\rangle & @\ \frac{1}{2} \\ |1\rangle & @\ \frac{1}{2} \end{cases}$$

For instance, in this example given $|0\rangle$ which represents the classical bit 0, we get a qubit as output of the linear transformation. In general, the operation performed by the Hadamard gate can be represented as follows:

$$\forall a \in \{0,1\} \quad \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^a\,|1\rangle\right)$$

As a side note, as we mentioned at the beginning of the chapter quantum mechanics has randomness intrinsically, and since the operation $H\,|0\rangle$ returns a qubit that has 50% of probability of being either $|0\rangle$ or $|1\rangle$ once measured, this operation provides a <u>true</u> random number generator.

Lastly, can we *represent* qubits graphically? Well, we may be tempted to anwer negatively to this question, since a qubit is described by two complex numbers $\alpha, \beta \in \mathbb{C}$, which implies that we actually need 4 dimensions to correctly represent our vector. However, through polar coordinates we can actually define a graphical representation which allows us to "picture" qubits, through the so called **Bloch sphere**. First, consider a qubit

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

for some $\alpha, \beta \in \mathbb{C}$ such that $|\alpha|^2 + |\beta|^2 = 1$, as usual. Now, recalling that any complex number $z \in \mathbb{C}$ can be actually written as follows

$$z = |z| e^{i\theta}$$

for some angle θ, we can actually rewrite our qubit as follows:

$$\begin{aligned}
|\psi\rangle &= |\alpha| e^{i\theta_\alpha} |0\rangle + |\beta| e^{i\theta_\beta} |1\rangle \\
&= e^{i\theta_\alpha} \left(|\alpha| |0\rangle + |\beta| e^{i\left(\theta_\beta - \theta_\alpha\right)} |1\rangle \right) \\
&= |\alpha| |0\rangle + |\beta| e^{i\left(\theta_\beta - \theta_\alpha\right)} |1\rangle & \left(e^{i\theta_\alpha} \text{ is a global phase}\right) \\
&= |\alpha| |0\rangle + e^{i\varphi} |1\rangle & (\text{let } \varphi := \theta_\beta - \theta_\alpha \in [0, 2\pi))
\end{aligned}$$

and finally, since $|\alpha|^2 + |\beta|^2 = 1$, is precisely the equation of the circumference of radius 1, we usually rewrite the last equation as follows:

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

where $\theta \in [0, \pi], \varphi \in [0, 2\pi)$. This formulation of the qubit $|\psi\rangle$ allows us to represent it inside the Bloch sphere: in fact, in this formulation the qubit is normalized, which implies that it will lie on a 3 dimensional unit sphere, and it is described by the two phases θ and φ — in 2D polar coordinates there is only 1 angle, as in 3D polar coordinate there are two angles.

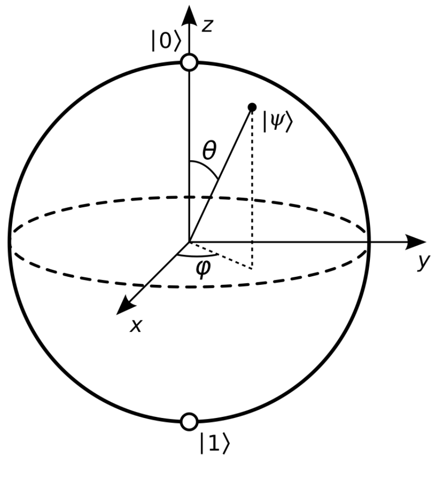

Figure 1.1: The Bloch sphere representing some qubit.

1.2.1 The tensor product

So far we have dealt with only one qubit at a time, but what if we have two qubits? First, let's look at the classical counterpart. If we take two bits $a, b \in \{0, 1\}$, we can represent 4 possible binary numbers, namely 00, 01, 10 and 11, which we can algebraically obtain by computing the usual cartesian product

$$\{0,1\}^2 = \{0,1\} \times \{0,1\} = \{(0,0),(0,1),(1,0),(1,1)\}$$

Note that in the cartesian products it holds that:

- the length of the tuples of the product is linear w.r.t. the number of factors of the cartesian products — in this case, 2
- each element of a tuple is *independent* from the other elements of the tuple

How can we evaluate all the possible states that two qubits can represent, instead? To answer this question, we need to introduce a new operator, which is called **tensor product**. Given two vectors $\begin{pmatrix} a \\ b \end{pmatrix}$ and $\begin{pmatrix} c \\ d \end{pmatrix}$, their tensor product is defined as follows

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} := \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}$$

Hence, consider two qubits

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \qquad |\phi\rangle = \beta |0\rangle + \beta_1 |1\rangle = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}$$

To obtain all the possible states of $|\psi\rangle$ and $|\phi\rangle$ we just have to compute the tensor product between them, which is

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \\ &= \alpha_0\beta_0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \alpha_0\beta_1 \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_1\beta_0 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \alpha_1\beta_1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

At the beginning of the chapter we defined $|0\rangle$ and $|1\rangle$ to be $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ without providing an explaination; now that we are dealing with more than 2 dimensions we can show why such names are used. In fact, we will use the following naming convention

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

and in general it holds that

$$|\text{bin}(i)\rangle = e_i$$

where $\text{bin}(i)$ represents for the binary representation of i, and e_i is the i-th vector of the canonical basis. This implies that we can rewrite the previous tensor product as follows:

$$|\psi\rangle \otimes |\phi\rangle = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle = \sum_{i,j\in\{0,1\}} \alpha_i\beta_j |ij\rangle$$

As a final note, it can be easily proven that

$$\forall i,j \in \{0,1\} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

For example, given two qubits

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

we get that

$$\begin{aligned} |\psi\rangle \otimes |\phi\rangle &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \otimes \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \\ &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ &\begin{cases} |0\rangle \text{ and } |0\rangle & @\ \frac{1}{4} \\ |0\rangle \text{ and } |1\rangle & @\ \frac{1}{4} \\ |1\rangle \text{ and } |0\rangle & @\ \frac{1}{4} \\ |1\rangle \text{ and } |1\rangle & @\ \frac{1}{4} \end{cases} \end{aligned}$$

where the probabilities at the end refer to the two individual qubits. To recap, in general the tensor product $|\psi\rangle \otimes |\phi\rangle$ of two qubits encodes the superposition of 4 basis states, namely $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$.

Moreover, the following property can be proved easily.

Proposition 1.1: Distributive property of $\otimes$

Given three qubits $|\psi\rangle, |\phi\rangle$ and $|\chi\rangle$, it holds that

$$(|\psi\rangle + |\phi\rangle) \otimes |\chi\rangle = |\psi\rangle \otimes |\chi\rangle + |\phi\rangle \otimes |\chi\rangle$$

1.2.2 Controlled operations

Another familyh of very important gates in quantum computing is the *controlled operations*. The first controlled operation that we are going to discuss is the so called **Controlled NOT (CNOT)** gate, which is defined as follows:

a	b	$\text{CNOT}(a, b)$
0	0	0
0	1	1
1	0	1
1	1	0

In fact, the names comes from the fact that the first input a is called *control bit*, which if set to 1 will flip the *target bit* b — in fact, in its implementation what actually happens is that b's wire itself is flipped. Therefore, in general we will write that

$$\text{CNOT}(a, b) = (a, a \oplus b)$$

First, we observe that this function is clearly not invertible, since for instance if we know that the output is 0 we still need the input a to evaluate if b was 0 or 1. Hence, to solve this issue we usually pair the output of CNOT with a itself, so that we can actually invert the computation.

Moreover, so far we only dealt with transformation that only expected one qubit argument as input, but the CNOT gate would certainly need 2 inputs to perform any computation, so how do we provide two inputs to it? As we showed before, we konw that

$$\forall i, j \in \{0, 1\} \quad |i\rangle \otimes |j\rangle = |ij\rangle$$

which directly implies that the vector $|ij\rangle$ encapsulated two qubits at once without ambiguity. Hence, we can actually leverage the tensor product to provide the input to the CNOT matrix, such that the quantum CNOT will behave as follows

$$\text{CNOT}(|a\rangle \otimes |b\rangle) = |a\rangle \otimes |a \oplus b\rangle$$

Hence, the matrix that behaves as such is the following

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

which expects a 4×1 input vector, and outputs a 4×1 output vector as well.

Laslty, as for the CNOT operator, we can actually define controlled operators for both Y and Z, which are respectively called CY and CZ operators.

1.2.3 Quantum circuits

Now that we introduced a couple of quantum gates, we can show how computation is actually represented in quantum computing. For instance, consider the following picture:

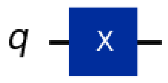

Figure 1.2: The NOT gate.

In this example, we have 1 single input qubit, namely q, and the box labeled with an X represents the NOT gate. We observe that, by convetion, all qubits in quantum circuits are assumed to be set to $|0\rangle$.

In the following example, instead, it is represented how the Hadamard gate looks like in quantum circuits.

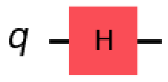

Figure 1.3: The Hadamard gate.

Moreover, if we consider two qubits as inputs q_0 and q_1, we can represent the CNOT operator as follows:

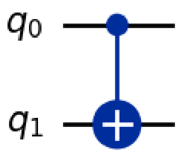

Figure 1.4: The CNOT gate.

We observe that q_1 then becomes the output of the CNOT operation, and q_0 remains unchanged. Lastly, the measurement operation is represented with the following picure:

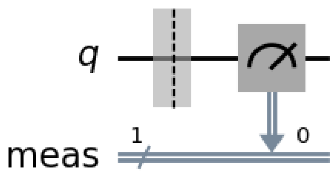

Figure 1.5: The measure operation.

In particular, in this cirtuit we see that:

- the vertical “double line” reprents *classical bits*
- the number 1 next to the label “meas” indicates the number of qubits that have been measured
- the number 0 is the index of the measured qubit

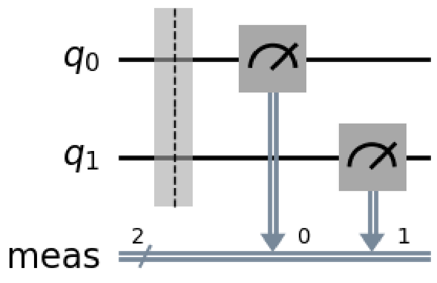

Figure 1.6: An exmaple of measurement of 2 qubits.

Lastly, another very important circuit is the following, which produces the so called **Greenberger-Horne-Zeilinger (GHZ)** state

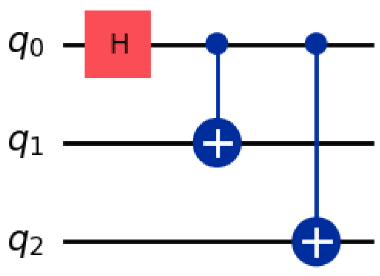

Figure 1.7: The GHZ quantum circuit.

which is represented as follows

$$|\mathrm{GHZ}\rangle := \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

1.3 Peculiarities of quantum mechanics

1.3.1 Quantum entanglement

Consider the following quantum state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Can this state be rewritten as the tensor product of two distinct quantum states? We observe that for this to be possible we would require some complex values $\alpha_0, \alpha_1, \beta_0, \beta_1$ such that

$$\begin{cases} \alpha_0\beta_0 = \alpha_1\beta_1 = 0 \\ \alpha_0\beta_1 = \alpha_1\beta_0 = \frac{1}{\sqrt{2}} \end{cases}$$

but $\alpha_0\beta_0 = 0$ implies that at least one between α_0 and β_0 has to be 0, meaning that at least one between $\alpha_0\beta_1$ and $\alpha_1\beta_0$ has to be 0 as well. This proves that there is no such pair of quantum states which can describe $|\psi\rangle$ through the tensor product operation. In fact, we see that

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \begin{cases} |01\rangle & @\ \frac{1}{2} \\ |10\rangle & @\ \frac{1}{2} \end{cases}$$

Indeed, this particular state we chose is one of the so called **Bell states**.

Definition 1.2: Bell states

The following are the four **Bell states**:

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

$$|\Phi^-\rangle := \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\Psi^+\rangle := \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\Psi^-\rangle := \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

Whenever we have a state $|\psi\rangle$ that cannot be represented as the tensor product of two simpler quantum states, we say that the state is **entangled** — or that its possible outcomes are entangled. In particular, entangled states describe a very weird phenomenon first proposed as a thought experiment in a groundbreaking paper by **Einstein, Podolsky and Rosen (EPR)** [EPR35], the so called **EPR paradox**.

The thought experiment involves a pair of particles prepared in such *entangled state*. Einstein, Podolsky, and Rosen pointed out that, in this state, if the position of the first particle were measured, the result of measuring the position of the second particle *could be predicted*. If instead the momentum of the first particle were measured, then the result of measuring the momentum of the second particle could be predicted. They argued that no action taken on the first particle could instantaneously affect the other, since this would involve information being transmitted faster than light, which is impossible according to the theory of relativity. Einstein famously called this phenomenon "spooky action at a distance", and to the best of our knowledge the theory of quantum mechanics says that if we have two engangled states, and measure one of them — for instance, say that it collapses to $|0\rangle$ — the other state will **instantaneously** collapse to $|1\rangle$ (and viceversa). They are *perfectly anti-correlated*, even if the two states are phisically light-years away from each other.

To be precise, entanglement is *not* a way to tranfer information — collapsing happens instantaneously, which would violate the fact that nothing can travel faster than light, not even information. Instead, it is a way to share correlations nonlocally. In fact, it is a phenomenon that regards the *whole quantum system* considered: for instance, given three qubits q_0, q_1, q_2, such that q_1 and q_2 are entangled, we might want to only measure $q_0 \otimes q_1$, which in turn will make q_2 collapse into some quantum state that has to be mathematically computed in order to be predicted — this will be more clear when we will describe **quantum teleportation** in Section 1.3.3.

To finish off this section, we can actually generate entangled states, or **EPR pairs** for short, through quantum gates as such:

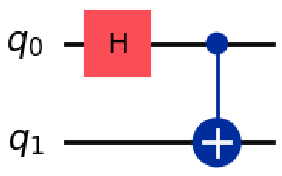

Figure 1.8: The quantum circuit for $|\Phi^+\rangle$.

In particular, we observe that the first Hadamard gate will transform $|0\rangle$ to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and through the CNOT operation we obtain

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

1.3.2 No-cloning theorem

An operation that we take for granted in classical computation is the possibility to *copy* the value of a bit: if Alice has two bits $x, y \in \{0, 1\}$, and she wants to copy the value of x into y, she can do it without any issues. However, in quantum mechanics this is *not* possible, because it would quite literally violate the laws of physics — as far as we understand it.

In 1982 Wootters and Zurek [WZ82] proved the so called **no-cloning theorem**, which states that it is impossible to create an independent and identical copy of an arbitrary *unknown* quantum state.

Theorem 1.1: No-cloning theorem

There is no quantum transformation that copies an unknown quantum state.

Proof. by way of contradiction, suppose that there exists such a transformation CP that is able to copy an unknown quantum state — and in particular, we observe that such transformation would have to be linear. But clearly, in order to have a copy we need to actually *store* it somewhere, so we can assume that CP has to take two inputs, one being the state that we want to copy and the other one being the state that we want to replace with the copy of the first one. In other words, we are assuming that

$$\exists y \forall x \quad \mathrm{CP}(x \otimes y) = x \otimes x$$

Now, through some algebraic manipulation we get that

$$\begin{aligned}
&\exists y \forall x \quad \mathrm{CP}(x \otimes y) = x \times x \\
\equiv &\exists y \forall x, a \quad \mathrm{CP}((x+a) \otimes y) = (x+a) \otimes (x+a) \\
\equiv &\exists y \forall x, a \quad \mathrm{CP}(x \otimes y + a \otimes y) = (x+a) \otimes (x+a) && \text{(by distributivity of } \otimes) \\
\equiv &\exists y \forall x, a \quad \mathrm{CP}(x \otimes y) + \mathrm{CP}(a \otimes y) = (x+a) \otimes (x+a) && \text{(by linearity of CP)} \\
\equiv &\exists y \forall x, a \quad x \otimes x + a \otimes a = x \otimes x + x \otimes a + a \otimes x + a \otimes a && \text{(by definition of CP)} \\
\equiv &\exists y \forall x, a \quad \mathbf{0} = x \otimes a + a \times x
\end{aligned}$$

which should be true for every x and every a, however it does not hold for $x = |0\rangle$ and $a = |1\rangle$, thus raising a contradiction $\lightning$. □

The no-cloning theorem represents an inherent limitation of quantum computation, and has direct impacts on **quantum cryptography** and **quantum error correction**, but must importantly it directly impacts a phenomenon called **quantum teleportation**

1.3.3 Quantum teleportation

So far we saw that quantum states cannot be cloned, but can we at least *send* them? Suppose that Alice wants to send Bob $|\psi\rangle$, described by some α and β. Clearly, the only thing that Bob has to receive are indeed the probability amplitudes of $|\psi\rangle$, so even if Alice cannot clone her quantum state, nothing prevents her to build a quantum circuit which *destroys* her $|\psi\rangle$ but does allow Bob to receive α and β. This process is called **quantum teleportation**, and can be achieved through the following algorithm.

Algorithm 1.1: Quantum teleportation algorithm

Given three qubits q_0, q_1 and q_2, the algorithm moves the state of q_0 into q_2

```
1:  function QuantumTeleportation($q_0$, $q_1$, $q_2$)
2:      $q_1 = H(q_1)$
3:      $q_2 = CX(q_1, q_2)$                                  ▷ entangle $q_1$ and $q_2$
4:      $q_1 = CX(q_0, q_1)$
5:      $q_0, q_1 = \text{measure}(q_0, q_1)$
6:      if $q_1 == |1\rangle$ then
7:          $q_2 = X(q_2)$
8:      end if
9:      if $q_0 == |1\rangle$ then
10:         $q_2 = Z(q_2)$
11:     end if
12:     return $q_2$
13: end function
```

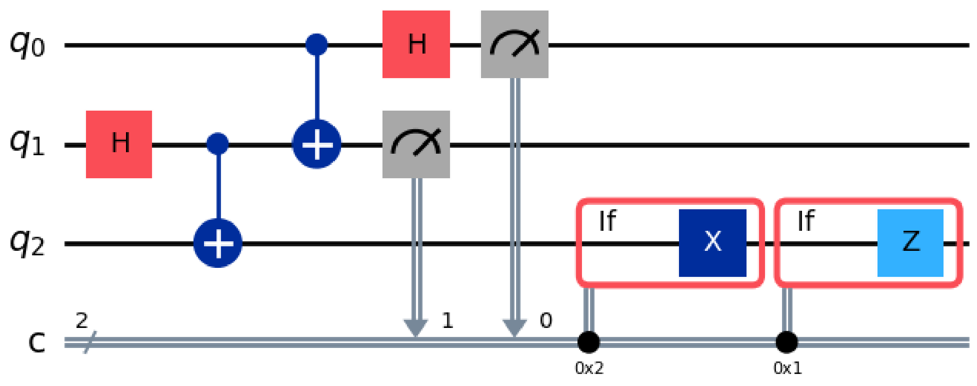

Figure 1.9: The Quantum Teleportation circuit.

There is quite a lot to unpack in this diagram. First, the quantum state that we want to teleport is q_0 in this diagram, and it will be teleported in q_2 at the end of the quantum computation.

In the first part of the circuit, we see that q_1 and q_2 are entangled (in an initial stage of the process, not performed by Alice nor Bob) in the Bell state $|\Phi^+\rangle$ thanks to the Hadamard and the CNOT gates — as we described in previous sections. In a real-world scenario, we will assume that q_1 and q_2 are given to Alice and Bob respectively (through some **quantum channel** such as optical fibers or free-space links in order to avoid *decoherence*), and quantum mechanics will guarantee that the teleportation will work even our two protagonists are thousands of kilometers away from each other.

After creating and entangling q_1 and q_2 (say for instance in a lab as preparation), we have the part of circuit that concerns Alice: in fact, she must apply a CNOT to her entangled qubit q_1, controlled by q_0, and then apply a Hadamard transformation to q_0. At this point, the circuit must apply a measurement to both q_0 and q_1 — and in particular, this operation will *destroy* the original state as previously anticipated.

Finally, it's Bob's turn: to obtain the original quantum state of q_0, the only thing he needs to do is first apply a CNOT to his entangled qubit q_2, controlled by q_1's outcome, followed by an application of a CZ, controlled by q_0's outcome instead — we observe that this part is indicated in the diagram through the `0x2` and `0x1` labels respectively. In fact, in the label `0xX` the number `X` represents the hexadecimal representation of the binary number obtained by joining the classical bits all together — for instance, in this circuit we have that 2 represents 10, meaning that only q_1 will be checked in the condition, and 1 represents 01, which means that only q_0 will be the control bit.

To show why the circuit actually works, we first need to discuss how computations with qubits and quantum gates is performed. In particular, we do not consider qubits *individually*, but instead we consider the whole **system** of qubits, i.e. $q_0 \otimes q_1 \otimes q_2$ simultaneously, and thus we will perform calculations as such. In fact, even if the drawing represents Alice's measurements of q_0 and q_1 independently, what happens in reality is that Alice is going to measure $q_0 \otimes q_1$ such that q_3 will collapse into its opposite.

Finally, we are ready to prove the correctness of the quantum teleportation circuit. First, we need to represent the initial quantum state, namely

$$\begin{aligned}
&q_0 \otimes q_1 \otimes q_2\\
=&\ket{\psi} \otimes \ket{\Phi^+}\\
=&(\alpha\ket{0}_0 + \beta\ket{1}_0) \otimes \frac{1}{\sqrt{2}}(\ket{00}_{12} + \ket{11}_{12})\\
=&\frac{1}{\sqrt{2}}[\alpha\ket{0}_0 \otimes (\ket{00}_{12} + \ket{11}_{12}) + \beta\ket{1}_0 \otimes (\ket{00}_{12} + \ket{11}_{12})]
\end{aligned}$$

where the notation $\ket{00}_{12}$ represents for example that we are considering q_1 and q_2's parts of states, respectively. From now on, we will omit the $\otimes$ symbol — as for the "normal" product. The next step is to apply the CNOT on q_0 and q_1, therefore the quantum state of the system becomes the following:

$$\frac{1}{\sqrt{2}}[\alpha\ket{0}_0(\ket{00}_{12} + \ket{11}_{12}) + \beta\ket{1}_0(\ket{10}_{12} + \ket{01}_{12})]$$

Next, we have to apply the Hadamard gate on q_0, which turns the quantum state into the following

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\left[\alpha\frac{1}{\sqrt{2}}(\ket{0}_0 + \ket{1}_0)(\ket{00}_{12} + \ket{11}_{12}) + \beta\frac{1}{\sqrt{2}}(\ket{0}_0 - \ket{1}_0)(\ket{10}_{12} + \ket{01}_{12})\right]\\
=&\frac{1}{2}[\alpha(\ket{000}_{012} + \ket{011}_{012} + \ket{100}_{012} + \ket{111}_{012}) + \beta(\ket{010}_{012} + \ket{001}_{012} - \ket{110}_{012} - \ket{101}_{012})]\\
=&\frac{1}{2}[\ket{00}_{01}(\alpha\ket{0}_2 + \beta\ket{1}_2) + \ket{01}_{01}(\beta\ket{0}_2 + \alpha\ket{1}_2) + \ket{10}_{01}(\alpha\ket{0}_2 - \beta\ket{1}_2) + \ket{11}_{01}(\alpha\ket{1}_2 - \beta\ket{0}_2)]\\
=&\frac{1}{2}[\ket{00}_{01}\ket{\psi}_2 + \ket{01}_{01}X\ket{\psi}_2 + \ket{10}_{01}Z\ket{\psi}_2 + \ket{11}_{01}XZ\ket{\psi}_2]
\end{aligned}$$

Finally, Alice will perform the measurement on q_0 and q_1, and what will happen is that the *whole* quantum state of the quantum circuit will collapse as follows:

$$\begin{cases} |00\rangle_{01} \otimes |\psi\rangle_2 & @\ \frac{1}{4} \\ |01\rangle_{01} \otimes X\,|\psi\rangle_2 & @\ \frac{1}{4} \\ |10\rangle_{01} \otimes Z\,|\psi\rangle_2 & @\ \frac{1}{4} \\ |11\rangle_{01} \otimes XZ\,|\psi\rangle_2 & @\ \frac{1}{4} \end{cases}$$

Note that to perform the measurement operation on just q_0 and q_1 we would need some mathematical tools that are outside the scope of this course, therefore we will only show the probabilities of the outcomes as presented above. In fact, fom this table we can easily explain the last part of the quantum teleportation circuit, i.e. Bob's part, as shown below.

Alice's outcome	Bob's part	Bob's result
0 and 0	I	$\|\psi\rangle$
0 and 1	X	$XX\,\|\psi\rangle = \|\psi\rangle$
1 and 0	Z	$ZZ\,\|\psi\rangle = \|\psi\rangle$
1 and 1	XZ	$XZXZ\,\|\psi\rangle = \|\psi\rangle$

Lastly, note that even if the mathematical calculations don't highlight the fact that q_0 and q_1 are measured, these two bits are effectively *destroyed* as we already described. In fact, q_2 will be the only usable qubit after the whole process — for instance, nothing prevents us from applying more transformations on the state $|00\rangle_{01} \otimes |\psi\rangle_2$ *mathematically*, but in reality q_0 and q_1 are not usable anymore.

2

Mathematical foundations

Now that we laid down some preliminary concepts regarding quantum machanics and quantum computaions, we need to discuss some **mathematical foundations** in order to progress and achieve a deeper meaning of the tools that we are going to use. In fact, at the end of this chapter we will state precisely the postulates of quantum mechanics, but in order to understand them we need to introduce some crucial definitions.

We will start our mathematical discussion with the definition of **scalar product** — we will assume the definitions of vector space, basis and linear independence are already known by the reader.

Definition 2.1: Scalar product

Given a scalar product vector space V, a **scalar product** $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{C} : (v, w) \mapsto \langle v, w \rangle$ is a function that satisfies the following properties:

- $\forall u, v, w \in V, \alpha, \beta \in \mathbb{C} \quad \langle u, \alpha v + \beta w \rangle = \alpha \langle u|v \rangle + \beta \langle u|w \rangle$
- $\forall u, v \in V \quad \overline{\langle u, v \rangle} = \langle v, u \rangle$ — where $\overline{z}$ is the conjugate of $z \in \mathbb{C}$
- $\forall u \in U \quad \langle u, u \rangle \geq 0$ and $\langle u, u \rangle = 0$ if and only if $u = 0$

Scalar products are also called *inner product*, and are used to define many other tools on top of the vector space considered.

Proposition 2.1

For any scalar product vector space V, any scalar product satisfies the following property:

$$\forall u, v, w \in V, \alpha, \beta \in \mathbb{C} \quad \langle \alpha u + \beta v, w \rangle = \overline{\alpha} \langle u, w \rangle + \overline{\beta} \langle v, w \rangle$$

Proof. TODO □ TODO

In particular, the scalar product that we are going to use for our purposes is defined as follows:

$$\forall u, v \in \mathbb{C}^n \quad \langle u, v\rangle = \sum_{i=1}^{n} \overline{u_i} v_i$$

In fact, we can prove that this is indeed a scalar product as follows:

- TODO TODO
- TODO TODO
- TODO TODO

From now on, when we refer to a "scalar product" we will refer to this particular definition.

We are finally ready to explain the "braket" that we used from the beginning of the previous chapter. This notation was invented by the Nobel Prize in Physics Paul Dirac, and it works as follows: first, observe that our scalar product can be rewritten as follows

$$\langle u, v\rangle = \begin{pmatrix} \overline{u_1} \cdots \overline{u_n} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

To be precise, this product would yield a 1×1 matrix, which can be interpreted as a scalar. Through Dirac notation, we will write

$$\langle u, v\rangle = \langle u|v\rangle$$

where $\langle u|$ is called **bra**, and $|v\rangle$ is called **ket** (as in "bra-ket"). In other words, we have that $|v\rangle$ is just a regular column vector $v \in V$

$$|v\rangle = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

defined over some scalar product vector space V, while $\langle u|$ is a *linear map* that acts as follows:

$$\langle\cdot| : V \to \overline{V} : \begin{pmatrix} u_1 \cdots u_n \end{pmatrix} \mapsto \begin{pmatrix} \overline{u_1} \cdots \overline{u_n} \end{pmatrix}$$

Theorem 2.1: Cauchy-Schwarz inequality

Given a scalar product vector space V, it holds that

$$\forall u, v \in V \quad |\langle u|v\rangle| \le \sqrt{\langle u|v\rangle \langle u|v\rangle}$$

where the equality holds if and only if u and v are linearly independent.

Moreover, our scalar product induces a **norm**, which is defined as follows.

Definition 2.2: Norm

Given a scalar product vector space V, the **norm** of a vector $v \in V$ is defined as follows

$$||v|| = \sqrt{\langle v|v\rangle}$$

As usual, two vectors $u, v \in V$ are said to be **orthogonal** if $\langle u|v\rangle = 0$. This allows us to define orthonormal bases.

Definition 2.3: Orthonormal basis

Given a scalar product vector space V, a basis $\{e_1, \dots, e_n\}$ is said to be **orthonormal** if

$$\forall i, j \in [n] \quad \langle e_i|e_j\rangle = \delta_{ij}$$

where $\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$ is called **Kronecker delta**.

Let's see the Dirac notation in action. Consider an orthonormal basis $\{e_1, \dots, e_n\}$ for some scalar product vector space V; by definition, we know that we can write any vector $u \in V$ as follows

$$u = \sum_{i=1}^{n} \alpha_i e_i$$

for some coefficients $\alpha_1, \dots, \alpha_n \in \mathbb{C}$. Now, we observe that for all $i \in [n]$

$$\begin{aligned} \langle e_i|u\rangle &= \langle e_i|\sum_{j=1}^{n} \alpha_j e_j\rangle \\ &= \langle e_i|\alpha_1 e_1 + \dots + \alpha_n e_n\rangle \\ &= \alpha_1 \langle e_i|e_1\rangle + \dots + \alpha_n \langle e_i|e_n\rangle \\ &= \sum_{j=1}^{n} \alpha_j \langle e_i|e_j\rangle \\ &= \sum_{j=1}^{n} \alpha_j \delta_{ij} \\ &= \alpha_i \end{aligned}$$

Indeed, with the scalar product we can compute the projection of u onto the i-th vector of the basis. Hence, we can rewrite the first equation as follows:

$$|u\rangle = \sum_{i=1}^{n} \alpha_i |e_i\rangle = \sum_{i=1}^{n} \langle e_i|u\rangle |e_i\rangle$$

In particular, we observe that

$$|u\rangle = \sum_{i=1}^{n} |e_i\rangle \langle e_i|u\rangle \implies I = \sum_{i=1}^{n} |e_i\rangle \langle e_i|$$

which is a famous identity in quantum mechanics called **resolution of the identity**. As a final note, by the properties of scalar products we also have that

$$\langle v|u\rangle = \left\langle v \middle| \sum_{i=1}^{n} \langle e_i|u\rangle |e_i\rangle \right\rangle = \sum_{i=1}^{n} \langle v|e_i\rangle \langle e_i|u\rangle$$

Proposition 2.2

Given a scalar product vector space V, it holds that

$$\forall u, v, w \in V \quad \langle u|v\rangle |w\rangle = (|w\rangle \langle u|) |v\rangle$$

Proof. TODO □ TODO

2.1 Hilbert spaces

Now that we covered Dirac notation, we can describe what are **Hilbert spaces** — we will see why we care about this particular type of vector spaces later in the chapter. First, consider the following definitions.

Definition 2.4: Weak convergence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence **converges weakly** to a vector $v \in V$ if

$$\forall w \in V \quad \lim_{m\to+\infty} \langle v_m|w\rangle = \langle v|w\rangle$$

In other words, this type ocf convergence requires all projections of v_m along any fixed direction w to approach the projection of v. Differently, the next type of convergence is more strict.

Definition 2.5: Strong convergence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence **converges strongly** to a vector $v \in V$ if

$$\lim_{m\to+\infty} ||v - v_m|| = 0$$

In fact, this type of convergence requires the actual vectors of the sequence to get close *in norm* to v. We observe the following proposition.

Proposition 2.3

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, if the sequence converges strongly to some vector $v \in V$, it holds that

- the sequence also converges weakly
- the scalar products defined over V are continuous, i.e.

$$\forall u, v \in V \quad \langle u|v\rangle = \lim_{m\to+\infty} \langle u|v_m\rangle$$

Definition 2.6: Cauchy sequence

Given a scalar product vector space V, and a vector sequence $\{v_m\}_{m\in\mathbb{N}}$ defined over V, we say that the sequence is a **Cauchy sequence** if it holds that

$$\forall\varepsilon > 0 \quad \exists n_\varepsilon \in \mathbb{N} \quad \forall n, m > n_\varepsilon \quad ||v_n - v_m|| < \varepsilon$$

For example, let's consider the space $\mathbb{R}^2$ equipped with the Euclidean norm

$$||v|| = \sqrt{x^2 + y^2}$$

Then, if we consider the following vector sequence

$$\left\{\begin{pmatrix} \frac{1}{m} \\ \vdots \\ \frac{1}{m} \end{pmatrix}\right\}_{m\in\mathbb{N}}$$

we see that for any distinct m, n it holds that

$$||v_m - v_n|| = \sqrt{\left(\frac{1}{m} - \frac{1}{n}\right)^2 + \left(\frac{1}{m} - \frac{1}{n}\right)^2} = \sqrt{2}\left|\frac{1}{m} - \frac{1}{n}\right|$$

Therefore, for any $\varepsilon > 0$ it suffices to take any $N > \dfrac{2\sqrt{2}}{\varepsilon}$ such that

$$\forall m, n > N \quad ||v_m - v_n|| < \varepsilon$$

We are finally ready to define Hilbert spaces.

Definition 2.7: Hilbert space

A **Hilbert space** is a *complete* scalar product vector space, i.e. it is a vector space

- equipped with a scalar product
- such that every Cauchy sequence converges strongly to an element in the space.

For example, the space $\mathbb{R}^n$ is a Hilbert space. Indeed, since every finite vector space of size n is isomporphic to $\mathbb{R}^n$, we can immediately derive the following proposition.

Proposition 2.4

Finite-dimensional vector spaces are always complete.

2.1.1 Linear operators

Given a Hilbert space $\mathcal{H}$, we can define **operators** — which are nothing but linear maps.

Definition 2.8: Adjoint operator

Given a Hilbert space $\mathcal{H}$, and an operator A, the **adjoint** operator of A, denoted with $A^\dagger$, is a linear map that satisfies the following property

$$\forall u, v \in \mathcal{H} \quad \langle u|A^\dagger v\rangle = \langle Au|v\rangle$$

We say that an operator A is **self-adjoint**, or *Hermitian*, if and only if $A = A^\dagger$.

For instance, the following matrix $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ is a linear operator whose adjoint is $S^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix}$. In fact, we have that

$$\langle u|S^\dagger v\rangle = \begin{pmatrix} \overline{u_1} & \overline{u_2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \overline{u_1} & \overline{u_2} \end{pmatrix} \begin{pmatrix} v_1 \\ -iv_2 \end{pmatrix} = \overline{u_1}v_1 - i\overline{u_2}v_2$$

and since

$$Su = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ iu_2 \end{pmatrix} \implies \langle Su| = \begin{pmatrix} \overline{u_1} & \overline{iu_2} \end{pmatrix}$$

but because $\overline{iu_2} = \overline{i} \cdot \overline{u_2} = -i\overline{u_2}$ this implies that

$$\langle Su|v\rangle = \begin{pmatrix} \overline{u_1} & -i\overline{u_2} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \overline{u_1}v_2 - i\overline{u_2}v_2$$

Proposition 2.5

For any adjoint operators A, B defined over some Hilbert space $\mathcal{H}$, it holds that

1. $(AB)^\dagger = B^\dagger A^\dagger$
2. for any scalar z it holds that $(zA)^\dagger = \overline{z}A^\dagger$
3. $(A^\dagger)^\dagger = A$
4. $(A + B)^\dagger = A^\dagger + B^\dagger$

How do we evaluate the adjoint of a given operator?

Proposition 2.6

Given an operator A defined over a scalar product vector space, it holds that

$$A^\dagger_{ij} = \overline{A_{ji}}$$

Proposition 2.7

If an operator A is self-adjoint, it holds that

$$\langle u|Av\rangle = \langle Au|v\rangle = \overline{\langle v|Au\rangle}$$

Proof. TODO □ TODO

At the beginning of the previous chapter we said that all quantum gates are *unitary transformations*, but we did now provide a definition of unitary. Now we are ready to introduce it, and start to grasp why we are discussing Hilbert spaces.

Definition 2.9: Unitary operators

Given a Hilbert space $\mathcal{H}$, and an operator U, we say that U is **unitary** if it holds that

$$UU^\dagger = U^\dagger U = I$$

In other words, U is unitary if and only if its adjoint operator is also its inverse. An interesting characterization of unitary transformations is the following property.

Proposition 2.8: Unitary operators (alt. def.)

An operator U defined over a Hilbert space $\mathcal{H}$ is unitary if and only if

- U is surjective
- $\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$ or equivalently, if it holds that

$$\forall x \in \mathcal{H} \quad ||Ux|| = ||x||$$

Proof. TODO □ TODO

In particular, we observe that the second property of this proposition is very interesting: the *preservation of scalar product*, i.e. the property for which

$$\forall x, y \in \mathcal{H} \quad \langle Ux|Uy\rangle = \langle x|y\rangle$$

means that the operator U does not change the geometric relationships between vectors — i.e. their lengths and angles remain the same.

Definition 2.10: Normal operators

Given a Hilbert space $\mathcal{H}$, and an operator A, we say that A is **normal** if it satisfies the following property

$$A^\dagger A = AA^\dagger$$

Clearly, from their definition we immediately see that both self-adjoint and unitary operators are both normal.

2.2 Spectral theory

Since unitary operators are linear maps, we are interested in their eigenvectors and eigenvalues — which are defined as usual. First, let us recall some preliminary definitions.

Definition 2.11: Non-degenerate eigenvalue

Given a matrix A, and an eigenvalue λ of A, we say that λ is **non-degenerate** if the associated eigenspace has dimension 1 (or equivalently, if it has only 1 associated eigenvector).

Definition 2.12: d-fold degenerate eigenvalue

Given a matrix A, and an eigenvalue λ of A, we say that λ is d-fold degenerate if there are d linearly independent eigenvectors $u_1, \ldots, u_d$ associated to λ.

In Dirac notation, if λ is non-degenerate, we refer to the only eigenvector associated to λ as $|\lambda\rangle$, indeed it holds that

$$A|\lambda\rangle = \lambda|\lambda\rangle$$

Proposition 2.9

Given a matrix A defined over a Hilbert space $\mathcal{H}$, and an eigenvalue λ associated to A, it holds that

$$\langle\lambda| A^\dagger = \overline{\lambda}\langle\lambda|$$

Proof. TODO □ TODO

The following theorem provides a characterization of the eigenvalues and eigenvectors of self-adjoint and unitary operators, which have suprisingly nice properties.

Theorem 2.2: Spectral theorem

The following propositions hold:

- The eigenvalues of a self-adjoint operator are real values.
- The eigenvalues of a unitary operator are complex values of modulus 1.
- Eigenvectors of self-adjoint and unitary operators associated to different eigenvalues are orthogonal to each other.

Moreover, for finite-dimensional Hilbert spaces the following holds.

Theorem 2.3: Spectral theorem for fin. Hilbert spaces

Given a finite-dimensional Hilbert space $\mathcal{H}$, and a normal operator A defined over $\mathcal{H}$, the set of all eigenvectors of A is an orthonormal basis for $\mathcal{H}$.

We can rewrite this thorem as follows: if we denote with u_{ij} the j-th eigenvector associated to the i-th eigenvalue of A, it holds that

$$\forall v \in \mathcal{H} \quad v = \sum_{i=1}^{m} \sum_{j=1}^{d_i} \alpha_{ij} u_{ij}$$

where d_i is the geometric multiplicity of the i-th eigenvalue of A. Indeed, we observe that $\dim \mathcal{H} = \sum_{i=1}^{m} d_i$. Lastly, through the Dirac notation we can rewrite the formula as follows

$$\forall v \in \mathcal{H} \quad |v\rangle = \sum_{i=1}^{m} \sum_{j=1}^{d_i} \langle \lambda_{ij} | v \rangle \, |\lambda_{ij}\rangle$$

2.3 Projectors

Next, we are going to discuss **projectors**, which are another very crucial pieces of quantum computation. We saw how scalar products are able to perform projection over desired direction, in fact we will use the Dirac notation to define precise operators for our purposes. But as always, first some preliminary definitions.

Definition 2.13: Orthogonal space

Given a scalar product vector space U, and two linear subspaces $V, W \subset U$, we say that V is orthogonal to W if

$$\forall v \in V, w \in W \quad \langle v | w \rangle = 0$$

Given a scalar product vector space U, and a linear subspace $V \subset U$, the **orthogonal**

complement of V is defined as follows:

$$V^{\perp} := \{u \in U \mid \forall v \in v \quad \langle u|v\rangle = 0\}$$

In particular, we observe that if U is finite-dimentional it holds that $V = U - V^{\perp}$ and that $(V^{\perp})^{\perp} = V$.

Definition 2.14: Topologically closed subspace

Given a Hilbert space $\mathcal{H}$, and a linear subspace $V \subset \mathcal{H}$, we say that V is **topologically closed** if any sequence of vectors defined over V converges in V.

Interestingly enough, given a topologically closed subspace $V \subset \mathcal{H}$ of some Hilbert space, we can write any vector $u \in V$ as the sum of two orthogonal vectors of V and $V^{\perp}$, as follows. Let $\{f_1, \dots, f_n\}$ be an orthonormal basis of V, and define the following vectors

$$\forall u \in U \quad u_V := \sum_{i=1}^{n} \langle f_i|v\rangle\, f_i$$

Then, if we call

$$u_{V^{\perp}} := u - u_V$$

, we see that TODO which indeed proves that u_V and $u_{V^{\perp}}$ are orthogonal to each other.

decommenta e finisci la formula

With this observation, we can finally define the projector operators.

Definition 2.15: Projector

Given a Hilbert space $\mathcal{H}$, and a closed subspace $V \subset \mathcal{H}$, the **projector** operator that projects any given vector $v \in \mathcal{H}$ onto V is defined as follows:

$$P_V : \mathcal{H} \to V : u \mapsto u_V = \sum_{i=1}^{n} \langle f_i|u\rangle\, f_i$$

where $\{f_1, \dots, f_n\}$ is an orthonormal basis of V.

In other words, the projector we have that

$$P_V := \sum_{i=1}^{n} |f_i\rangle \langle f_i|$$

Clearly, by definition of $u_{V^{\perp}}$ it holds that

$$P_{V^{\perp}} : \mathcal{H} \to V^{\perp} : u \mapsto u_{V^{\perp}} := u - u_V$$

Moreover, since P_V performs a projection, we have that

$$\forall u \in \mathcal{H} \quad u \in V \iff P_V u = u$$

and that

$$\forall u \in \mathcal{H} \quad u \in V^{\perp} \iff P_V u = 0$$

Additionally, for any projector P_V it holds that $P_V^2 = P_V$, by idempotency, and $P_V^{\dagger} = P_V$.

Proposition 2.10

Any projector operator only has 0 and 1 as possible eigenvalues.

Proof. Consider a projector operator P_V; by definition v is an eigenvalue of P_V associated to the eigenvalue λ if it holds that

$$P_V v = \lambda v$$

Hence, we observe that

$$P_V^2 v = P_V(P_V v) = P_V(\lambda v) = \lambda P_V v = \lambda(\lambda v) = \lambda^2 v$$

and by idempotency of P_V it holds that

$$\lambda^2 v = P_V^2 = P_V v = \lambda v$$

which implies that

$$\lambda^2 v - \lambda v = 0 \iff (\lambda^2 - \lambda)v = 0$$

Finally, since v is an eigenvector it holds that $v \neq 0$, therefore it must be that the last equation is true only if

$$\lambda^2 - \lambda = 0 \iff \lambda = 0 \lor \lambda = 1$$

□

Given a Hilbert space $\mathcal{H}$, and two topologically closed subspaces $V, W \subset \mathcal{H}$, we say that P_V and P_W are **orthogonal** if it holds that $V \perp W$. Now, fix a vector $u \in \mathcal{H}$; by definition $P_V u$ is a vector that lies inside V, therefore it holds that

$$P_W(P_V u) = 0$$

since $V \perp W$, and by the same reasoning applied on W first we conclude that

$$P_W P_V = P_V P_W = \mathbf{0}$$

where $\mathbf{0}$ is the **zero operator** — indeed, it is a zero matrix.

As a final note, if A is an linear operator, and λ is an eigenvalue of A, we denote with P_λ the projector that projects vectors onto the eigenspace associated to λ.

Proposition 2.11

If P and Q are two orthogonal projectors, then $P + Q$ is still a projector.

Proof. TODO □ TODO

2.4 Tensor product

Finally, the last ingredient that we need to discuss is the **tensor product**

TODO ________ TODO

2.5 Rules of quantum mechanics

Now that we defined Hilbert spaces and their operators in great detail, we can finally present we needed this mathematical foundations in order to progress: quantum mechanics is developed over Hilbert spaces with *countable* bases, and quantum computing works with finite-dimensional Hilbert spaces. In particular, there are four fundamental **postulates of quantum mechanics** which describe the : ________ what?

1. **State postulate**: the state of a quantum system is completely described by a vector $|\psi\rangle$ in a Hilbert space $\mathcal{H}$ — $|\psi\rangle$ is usually normalized, as we saw at the beginning of the previous chapter, and physical systems of different types live in different Hilbert spaces.
2. **Measurement postulate**: every measurable (i.e. *observable*) quantity corresponds to a self-adjoint operator on $\mathcal{H}$. In particular, given an observable A, and a state $v \in \mathcal{H}$, it holds that:
   - the only possible results of measuring A are one of its eigenvalues
   - the probability of measuring eigenvalue λ in state v is given by
   $$\Pr[A = \lambda \mid v] = \langle v | P_\lambda v \rangle$$
   where P_λ is the linear map that projects v onto the λ-eigenspace
3. **Time evolution postulate**: a closed system evolves through time according to the Schrödinger equation
   $$i\hbar \frac{\mathrm{d}}{\mathrm{d}t} v(t) = H v(t)$$
   where the elements of this first-order linear differential equation are the following:
   - $v(t)$ is the state vector at time t (a vector in a Hilbert space)
   - H is the system *Hamiltonian*, a self-adjoint operator that describes the total energy of the system
4. **Composite systems postulate**: the Hilbert space of a composite system is the tensor product of the Hilbert spaces of its subsystems. In other words, if system A is defined over $\mathcal{H}_A$, and system B is defined over $\mathcal{H}_B$, the total system lives in
   $$\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$$

If we take a closer look at the second postulate, we can actually explain why we choose that particular scalar product to be the probability. Since by convention any quantum state is normalize, i.e. $||v|| = 1$, it holds that

$$\begin{aligned} 1 &= ||v||^2 \\ &= \langle v|v\rangle \\ &= \left\langle \sum_{i=1}^{m} P_{\lambda_i v} \middle| \sum_{j=1}^{m} P_{\lambda_j v} \right\rangle \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} \langle P_{\lambda_i} v | P_{\lambda_j} v\rangle \end{aligned}$$

Now, since each P_{λ_i} is a projector, we know that when $i \neq j$ it holds that $P_{\lambda_i}P_{\lambda_j} = \mathbf{0}$, therefore by self-adjointness of projectors we have that

- if $i \neq j$ then
$$\langle P_{\lambda_i} v | P_{\lambda_j} v\rangle = \langle v | P_{\lambda_i} P_{\lambda_j} v\rangle = \langle v|\mathbf{0}v\rangle = 0$$
- if $i = j$ then
$$\langle P_{\lambda_i} v | P_{\lambda_j} v\rangle = \langle P_{\lambda_i} v | P_{\lambda_i} v\rangle = ||P_{\lambda_i} v||^2$$

Therefore, by adding only the non-zero terms we get that

$$\sum_{i=1}^{m} ||P_{\lambda_i} v||^2 = 1$$

Hence, we define

$$\Pr[A = \lambda_i \mid v] := ||P_{\lambda_i} v||^2$$

such that

$$\Pr[A \mid v] = \sum_{i=1}^{m} \Pr[A = \lambda_i \mid v] = \sum_{i=1}^{m} ||P_{\lambda_i} v||^2 = ||v||^2 = 1$$

which also means that our probabilities will add up to 1 automatically. Finally, we can rewrite this proability as follows (we will drop the index of the eigenvalue):

$$\begin{aligned} \Pr[A = \lambda \mid v] &= \langle P_\lambda v | P_\lambda v\rangle && \text{(by def. of adjoint)} \\ &= \langle v | P_\lambda^\dagger P_\lambda v\rangle && \text{(by self-adjointness)} \\ &= \langle v | P_\lambda^2 v\rangle && \text{(by idempotency)} \\ &= \langle v | P_\lambda P_\lambda v\rangle \\ &= \langle v | P_\lambda v\rangle \end{aligned}$$

Another postulate that we need to discuss is the third one, regarding the Schrödinger equation. In particular, the solution of the latter is

$$v(t_1) = U(t_2, t_1)v(t_1)$$

where $U(t_2, t_1)$ is called **time-evaluation operator**, and it is defined as follows:

$$U(t_2, t_1) = e^{-\frac{i}{\hbar}H(t_2 - t_1)}$$

(assuming H does not depend on time). We recall that H is a matrix, so we are raising e to the power of a matrix, an operation that is defined by the power series of the exponential as follows:

$$e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!}$$

What is interesting about this operator is that U is **unitary**, and in order to show it is suffices to prove that $U^\dagger = U^{-1}$. But how do we compute the adjoint of U? We observe that by the properties of the adjoint operation it holds that

$$\left(e^A\right)^\dagger = \left(\sum_{n=0}^{\infty} \frac{A^n}{n!}\right)^\dagger = \sum_{n=0}^{\infty} \frac{(A^n)^\dagger}{n!} = \sum_{n=0}^{\infty} \frac{(A^\dagger)^n}{n!} = e^{A^\dagger}$$

which means that the adjoint of an exponential is the exponential of the adjoint. This suffices to prove that

$$U^\dagger = \left(e^{-\frac{i}{\hbar}H(t_2-t_1)}\right)^\dagger = e^{\left(-\frac{i}{\hbar}H(t_2-t_1)\right)^\dagger} = e^{\frac{i}{\hbar}H(t_2-t_1)} = \left(e^{-\frac{i}{\hbar}H(t_2-t_1)}\right)^{-1} = U^{-1}$$

This is a crucial characteristic for quantum mechanics: since U is unitary, we know that it preserves the scalar product, therefore it also preserve **probabilities and norms**. This is why we say that evolution in quantum systems — or *quantum evolution*, for short — is unitary.

3
Quantum algorithms

TODO

non so di che parlerà il capitolo

When we introduced quantum operators we underlined the fact that each quantum gate has to be a *unitary* operator, and we now have the mathematical foundation to know that if a matrix is unitary, it is clearly also invertible — indeed, its adjoint is its inverse. This directly implies a very important property of quantum computation: except for the measurement operation, every quantum computation operation is **reversible**.

Truth be precise, classical computation *admits* reversible computation. In fact, we observe that we do have examples of reversible classical computaion that does not lose efficiency — in terms of time. In 1963 Lecerf and Euratom (Bruxelles) [LEB64] proposed a reversible Turing machine, and in 1973 a landmark result by Bennett [Ben73] proved that any standard Turing machine can be actually simulated by a reversible one — his construction involves augmenting the Turing machine with an auxiliary *history tape*, which can potentially lead to a large space overhead.

Consider the following bitwise operator T, that given three bits it works as follows:

$$T : \{0,1\}^3 \to \{0,1\}^3 : (a,b,c) \mapsto (a,b,c \oplus (a \wedge b))$$

The corresponding classical gate of this bitwise operator is called **Toffoli gate**, and it has very special characteristics. First, observe how it computes: while a and b remains unchanged, c is basically flipped if and only if both a and b are set to 1. In other words, both a and b act as "control bits"over the "target bit" c, indeed the Toffoli gate is sometimes called **Controlled Controlled NOT (CCNOT)**. As we did for the CNOT, this operator is clearly invertible since we are passing the bits a and b to the output too — also, the truth table of the Toffoli gate easily shows that it is indeed invertible. Moreover, we observe that by associativity of the XOR it holds that

$$(c \oplus (a \wedge b)) \oplus (a \wedge b) = c \oplus (a \wedge b) \oplus (a \wedge b) = c$$

which directly implies that

$$T^2 = I \implies T = T^{-1}$$

However, above all the most important property of the Toffoli gate is that it is **universal**. In fact, even though gates like NAND and NOR are universal too, they are *irreversible*. This implies that with the T operator we can build any reversible Boolean function, and since any ordinary Boolean function can be embedded into a reversible one — by adding extra bits to make it invertible — any classical computation can be simulated using Toffoli gates only.

3.1 Deutsch's algorithm

Even though quantum computation provides reversibility "for free", we saw that classical computation can still achieve invertibility of computation. However, the next characteristic that we are going to describe has no classical analogue.

First, let's start with a problem seemingly unrelated to our discussion. Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we would like to embed f inside a quantum computation. However, when $n \geq 3$ we are guaranteed that $f(x)$ is not reversible — it cannot be injective. This is a problem, since in quantum coputing all gates must be reversible — given that quantum evolution is unitary. Thus, how do we turn f into a reversible computation?

We define a map U_f defined as follows:

$$U_f : \{0,1\}^{n+1} \to \{0,1\}^{n+1} : (x, y) \mapsto (x, y \oplus f(x))$$

First, we observe that

$$(y \oplus f(x)) \oplus f(x) = y \oplus (f(x) \oplus f(x)) = y$$

which trivially proves that U_f is reversible. Moreover, we can actually prove that when applied to qubits the corresponding quantum operator

$$U_f : \mathcal{H} \to \mathcal{H} : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$$

is indeed unitary — we observe that we are omitting the tensor product symbol in the function definition, as usual in the literature.

Proposition 3.1

Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, the operator U_f is unitary.

Proof. TODO $\square$

TODO

This proves that the construction of U_f is precisely the gate that allows us to embed f into any quantum computation. Moreover, we observe that

- $|y\rangle = |0\rangle \implies U_f\,|x\rangle\,|0\rangle = |x\rangle\,|f(x)\rangle$
- $|y\rangle = |1\rangle \implies U_f\,|x\rangle\,|1\rangle = |x\rangle\,|\neg f(x)\rangle$

However, until now we only considered already collapsed qubits, but what if we consider a quantum input that is in a superposition? For instance, let

$$|x\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

and assume that $|y\rangle = |0\rangle$ for simplicity; this implies that

$$\begin{aligned} U_f |x\rangle |y\rangle &= U_f \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= U_f \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle) \\ &= \frac{1}{\sqrt{2}}(U_f |00\rangle + U_f |10\rangle) \qquad \text{(by linearity of } U_f\text{)} \\ &= \frac{1}{\sqrt{2}}(|0\rangle |f(0)\rangle + |1\rangle |f(1)\rangle) \end{aligned}$$

But notice what just happened: both $f(0)$ and $f(1)$ have been computed **simultaneously**, in one gate application. This has no classical equivalent, we would have to evaluate $f(0)$ and $f(1)$ *separately*. This phenomenon is called **quantum parallelism**, and it can be achieved only because:

- qubits are in superpositions
- quantum gates are linear

However, we observe that the result of our calculations is *still a superposition*. In fact, if we measure the output of $U_f |x\rangle |y\rangle$ we would still get either $|0\rangle |f(0)\rangle$ or $|1\rangle |f(1)\rangle$, both with 50% probability. This is a problem: the fact that we can compute $f(0)$ and $f(1)$ at the same time seems promising, but can we retrieve their actual values?

Unfortunately, this is not possible. Indeed, quantum parallelism cannot help us with *local* properties — i.e. when we need all individual outputs — it can only help when we need **global** properties. This limit derives from the fact that measurements prevent "seeing" both outcomes, in fact if we were able to compute $f(0)$ and $f(1)$ simultaneously from this superposition we would be violating the laws of quantum mechanics themselves.

Then, how do we extract useful *global* information from the superposition output? In 1985 Deutsch [Deu85] defined a quantum algorithm which is able to compute $f(0) \oplus f(1)$, which clearly tells us if $f(0)$ equals $f(1)$ or not.

Algorithm 3.1: Deutsch algorithm

Given a Boolean function f and 2 qubits, the algorithm returns $|0\rangle$ if f if $f(0) = f(1)$, $|1\rangle$ otherwise.

```
1: function DEUTSCH($f$, $q_0$, $q_1$)
2:     $q_1 = X(q_1)$
3:     $q_0, q_1 = (H \otimes H)(q_0, q_1)$
4:     $q_0, q_1 = U_f(q_0, q_1)$
5:     $q_0 = H(q_0)$
6:     return measure($q_0$)
7: end function
```

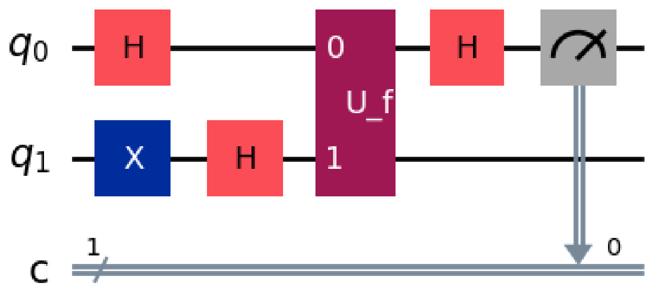

Figure 3.1: The quantum circuit for Deutsch's algorithm. The box colored in *magenta* labeled with `U_f` represents a "black-box" for whatever computation U_f represents (which directly depends on the chioce of f).

Proving that this quantum circuit is correct will be a little more involved than the quantum teleportation one. First, we need a lemma that will simplify our calculations.

Lemma 3.1

For any Boolean function f defined on n bits, and $a \in \{0, 1\}$, it holds that

$$U_f |a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = (-1)^{f(a)} |a\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Proof. First, by algebraic manipulation we see that

$$\begin{aligned}
U_f \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle) &= U_f \frac{1}{\sqrt{2}}(\left|a0\right\rangle - \left|a1\right\rangle) \\
&= \frac{1}{\sqrt{2}}(U_f \left|a0\right\rangle - \left|a1\right\rangle) \\
&= \frac{1}{\sqrt{2}}(\left|a\ f(a)\right\rangle - \left|a\ \neg f(a)\right\rangle)
\end{aligned}$$

and now, we observe that

- if $f(a) = 0$, then

$$\frac{1}{\sqrt{2}}(\left|a0\right\rangle - \left|a1\right\rangle) = (-1)^0 \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

- if $f(a) = 1$, then

$$\frac{1}{\sqrt{2}}(\left|a1\right\rangle - \left|a0\right\rangle) = (-1)^1 \left|a\right\rangle \otimes \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

□

We are now ready to prove the correctness of Deutsch's algorithm. To make things less cluttered, we will use the following standard notation:

$$\left|+\right\rangle := \frac{1}{\sqrt{2}}(\left|0\right\rangle + \left|1\right\rangle) \qquad \left|-\right\rangle := \frac{1}{\sqrt{2}}(\left|0\right\rangle - \left|1\right\rangle)$$

In particular, we observe that

$$H \left|0\right\rangle = \left|+\right\rangle \qquad H \left|1\right\rangle = \left|-\right\rangle$$

Moreover, we will omit the subscript of the corresponding qubit when the context is clear enough

$$\begin{aligned}
& q_0 \otimes q_1 \\
&= |0\rangle \otimes |0\rangle \\
\xrightarrow{X(q_1)} & |0\rangle \otimes |1\rangle \\
\xrightarrow{(H\otimes H)(q_0,q_1)} & |+\rangle \otimes |-\rangle \\
&= \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \\
&= \frac{1}{\sqrt{2}} |0\rangle_0 |-\rangle_1 + \frac{1}{\sqrt{2}} |1\rangle_0 |-\rangle_1 \\
\xrightarrow{U_f(q_0,q_1)} & \frac{1}{\sqrt{2}}(-1)^{f(0)} |0\rangle_0 |-\rangle_1 + \frac{1}{\sqrt{2}}(-1)^{f(1)} |1\rangle_0 |-\rangle_1 && \text{(by the lemma)} \\
&= \frac{1}{\sqrt{2}} \left((-1)^{f(0)} |0\rangle_0 + (-1)^{f(1)} |1\rangle_0\right) \otimes |-\rangle_1 \\
\xrightarrow{H(q_0)} & \frac{1}{\sqrt{2}} \left((-1)^{f(0)} |+\rangle_0 + (-1)^{f(1)} |-\rangle_0\right) \otimes \sqrt{2} |-\rangle_1 \\
&= \frac{1}{2} \left((-1)^{f(0)}(|0\rangle + |1\rangle) + (-1)^{f(1)}(|0\rangle - |1\rangle)\right) \otimes \sqrt{2} |-\rangle_1 \\
&= \frac{1}{2} \left(\left((-1)^{f(0)} + (-1)^{f(1)}\right) |0\rangle + \left((-1)^{f(0)} - (-1)^{f(1)}\right) |1\rangle\right) \otimes \sqrt{2} |-\rangle_1
\end{aligned}$$

Now, since the final operation of the circuit involves measuring $q_0 = \alpha |0\rangle + \beta |1\rangle$, the only two things that we care about are its probability amplitudes, namely

$$\alpha = \frac{1}{2} \left((-1)^{f(0)} + (-1)^{f(1)}\right)$$

$$\beta = \frac{1}{2} \left((-1)^{f(0)} + (-1)^{f(1)}\right)$$

and we see that

- if $f(0) = f(1)$, then

  $$(-1)^{f(0)} = (-1)^{f(1)} \implies \begin{cases} \alpha = \frac{1}{2} \left(2(-1)^{f(0)}\right) = (-1)^{f(0)} \\ \beta = 0 \end{cases}$$

  which implies that

  $$q_0 = (-1)^{f(0)} |0\rangle + 0 \cdot |1\rangle = (-1)^{f(0)} |0\rangle$$

  and we can ignore the $(-1)^{f(0)}$ factor since its a global phase

- if $f(0) \neq f(1)$, then

  $$(-1)^{f(0)} = -(-1)^{f(1)} \implies \begin{cases} \alpha = 0 \\ \beta = \frac{1}{2} \left(2(-1)^{f(0)}\right) = (-1)^{f(0)} \end{cases}$$

which implies that

$$q_0 = 0 \cdot |0\rangle + (-1)^{f(0)} |1\rangle = (-1)^{f(0)} |1\rangle$$

by the same reasoning as the other case

In the end, this proves that if $f(0) = f(1)$, q_0 will collapse to $|0\rangle$, while if $f(0) \neq f(1)$ q_1 will colapse to $|1\rangle$, proving that Deutsch's algorithm works correctly.

3.2 Deutsch-Josza algorithm

Even though Deutsch's algorithm is quite interesting and offers advantages that classical computation cannot achieve, still it seems like it wouldn't be very useful in practice. In fact, usually we are interested in the *values* of $f(0)$ and $f(1)$, and as we already mentioned quantum mechanics will not allow us to compute both the values at the same time — meaning that even if we use Deutsch's algorithm to now whether $f(0)$ is equal to $f(1)$ or not, we would still need to compute at least one between $f(0)$ and $f(1)$ in order to know both values.

This is because, in reality, the algorithm that we are using is only solving a particular case of a more complex problem. In fact, a couple of years later Deutsch [Deu92] realized that if we use $q_1 = |1\rangle$ and $q_0 = |0^n\rangle$ (i.e. we use n qubits set to $|0\rangle$) this algorithm is actually able to tell **constant** and **balanced** functions apart.

Definition 3.1: Constant function

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be **constant** if

$$\exists b \in \{0,1\} \quad \forall x \in \{0,1\}^n \quad f(x) = b$$

The definition of constant Boolean function has nothing special, and balanced functions are exactly what the name suggests, i.e. half of the inputs output 0 and the other half output 1, which can be succintly expressed as follows.

Definition 3.2: Balanced function

A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is said to be **balanced** if it holds that

$$\sum_{x \in \{0,1\}^n} f(x) = 2^{n-1}$$

We observe that a Boolean function can be neither constant nor balanced, so this decision problem is actually a **promise problem**: given a Boolean function f that is either constant or balanced — note that it cannot be both — decide if the function is constant or balanced. Indeed, we see that Deutsch's algorithm solved the same exact problem for $n = 2$: in fact, if $f(0) = f(1)$ it means that f is constant, otherwise the latter is balanced.

Moreover, this problem actually shows the power of quantum parallelism more evidently: with a classical computation, to solve this decision problem we would need at most

$$2^{n-1} + 1 = O(2^n)$$

queries to f, instead our quantum computation still only requires one evaluation of f to solve the problem.

Algorithm 3.2: Deutsch-Josza algorithm

Given a Boolean function f and $n+1$ qubits, the algorithm returns $|0^n\rangle$ if f is constant, $|1\rangle$ otherwise.

```
1: function DeutschJosza($f$, $q_0$, $q_1$)
2:     $q_1 = X(q_1)$
3:     $q_0, q_1 = (H^{\otimes n} \otimes H)(q_0, q_1)$
4:     $q_0, q_1 = U_f(q_0, q_1)$
5:     $q_0 = H^{\otimes n}(q_0)$
6:     return measure($q_0$)
7: end function
```

Note that in this algorithm q_0 are actually n qubits. Before proving the correctness of this general version of the algorithm, let us first take a look at the quantum circuit that defines it.

TODO

find a way to draw it

Bibliography

[Ben73] C. H. Bennett. “Logical Reversibility of Computation”. In: *IBM Journal of Research and Development* 17.6 (Nov. 1973), 525–532. ISSN: 0018-8646. DOI: 10.1147/rd.176.0525. URL: http://dx.doi.org/10.1147/rd.176.0525.

[Deu85] David Deutsch. “Quantum theory, the Church–Turing principle and the universal quantum computer”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 400.1818 (1985), pp. 97–117.

[Deu92] Josza Deutsch. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (Dec. 1992), 553–558. ISSN: 2053-9177. DOI: 10.1098/rspa.1992.0167. URL: http://dx.doi.org/10.1098/rspa.1992.0167.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. “Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?” In: *Physical Review* 47.10 (May 1935), 777–780. ISSN: 0031-899X. DOI: 10.1103/physrev.47.777. URL: http://dx.doi.org/10.1103/PhysRev.47.777.

[LEB64] Y. Lecerf and Communauté Européenne de l’énergie atomique Euratom (Bruxelles). *Logique mathématique: 1. Machines de Turing réversibles. Récursive insolubilité en n epsilon ny de l’équation u.* Euratom, 1964. URL: https://books.google.it/books?id=e1xjGwAACAAJ.

[WZ82] W. K. Wootters and W. H. Zurek. “A single quantum cannot be cloned”. In: *Nature* 299.5886 (Oct. 1982), 802–803. ISSN: 1476-4687. DOI: 10.1038/299802a0. URL: http://dx.doi.org/10.1038/299802a0.