SAPIENZA
UNIVERSITÀ DI ROMA

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

Quantum Computing

Author
Alessio Bandiera

November 14, 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 $o_1, \dots, o_N$, each occurring with an associated probability $p_1, \dots, p_N$ — 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 true 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.

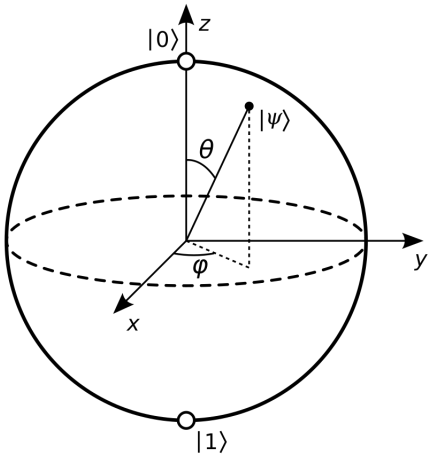


Figure 1.1: The Bloch sphere representing some qubit.

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	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.

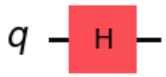


Figure 1.3: The Hadamard gate.

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

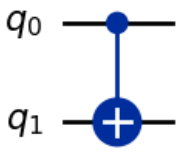


Figure 1.4: The CNOT gate.

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:

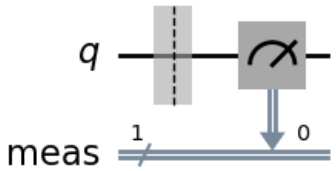


Figure 1.5: The measure operation.

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

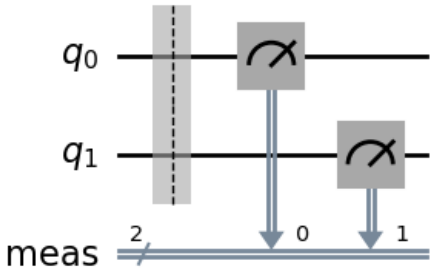


Figure 1.6: An exmaple of measurement of 2 qubits.

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

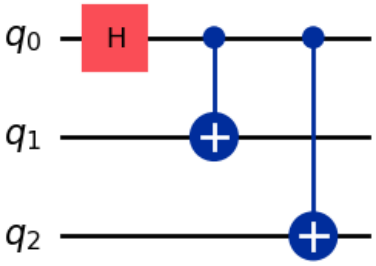


Figure 1.7: The GHZ quantum circuit.

which is represented as follows

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

1.3 Peculiarities of quantum mechanics

1.3.1 Quantum entanglement

Consider the following quantum state

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

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

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

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**:

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

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

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

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

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:

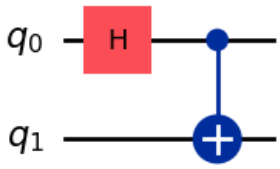


Figure 1.8: The quantum circuit for $|\Phi^+\rangle$.

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(q0, q1, q2)
2:     q1 = H(q1)
3:     q2 = CX(q1, q2)                          ▷ entangle q1 and q2
4:     q1 = CX(q0, q1)
5:     q0, q1 = measure(q0, q1)
6:     if q1 == |1⟩ then
7:         q2 = X(q2)
8:     end if
9:     if q0 == |1⟩ then
10:        q2 = Z(q2)
11:    end if
12:    return q2
13: end function
```

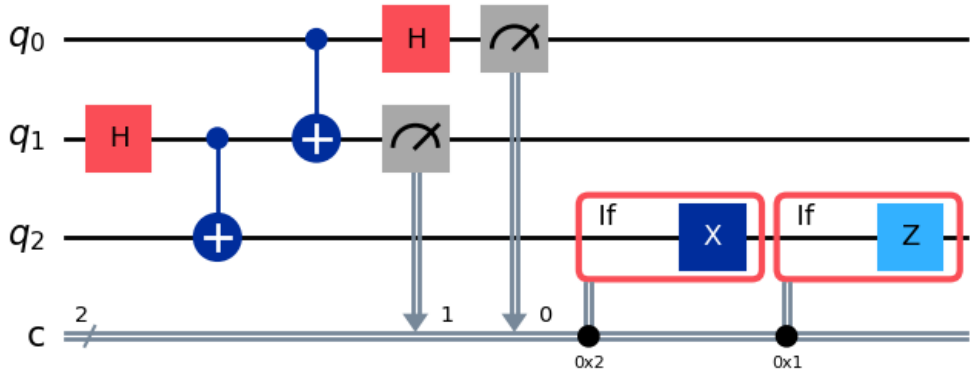


Figure 1.9: The Quantum Teleportation circuit.

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

$$\begin{aligned}
&q_0 \otimes q_1 \otimes q_2 \\
=&\left|\psi\right\rangle \otimes \left|\Phi^+\right\rangle \\
=&(\alpha \left|0\right\rangle_0 + \beta \left|1\right\rangle_0) \otimes \frac{1}{\sqrt{2}}(\left|00\right\rangle_{12} + \left|11\right\rangle_{12}) \\
=&\frac{1}{\sqrt{2}}[\alpha \left|0\right\rangle_0 \otimes (\left|00\right\rangle_{12} + \left|11\right\rangle_{12}) + \beta \left|1\right\rangle_0 \otimes (\left|00\right\rangle_{12} + \left|11\right\rangle_{12})]
\end{aligned}$$

where the notation $\left|00\right\rangle_{12}$ 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:

$$\frac{1}{\sqrt{2}}\left[\alpha \left|0\right\rangle_0 (\left|00\right\rangle_{12} + \left|11\right\rangle_{12}) + \beta \left|1\right\rangle_0 (\left|10\right\rangle_{12} + \left|01\right\rangle_{12})\right]$$

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

$$\begin{aligned}
&\frac{1}{\sqrt{2}}\left[\alpha\frac{1}{\sqrt{2}}(\left|0\right\rangle_0 + \left|1\right\rangle_0)(\left|00\right\rangle_{12} + \left|11\right\rangle_{12}) + \beta\frac{1}{\sqrt{2}}(\left|0\right\rangle_0 - \left|1\right\rangle_0)(\left|10\right\rangle_{12} + \left|01\right\rangle_{12})\right] \\
=&\frac{1}{2}\left[\alpha(\left|000\right\rangle_{012} + \left|011\right\rangle_{012} + \left|100\right\rangle_{012} + \left|111\right\rangle_{012}) + \beta(\left|010\right\rangle_{012} + \left|001\right\rangle_{012} - \left|110\right\rangle_{012} - \left|101\right\rangle_{012})\right] \\
=&\frac{1}{2}\left[\left|00\right\rangle_{01}(\alpha \left|0\right\rangle_2 + \beta \left|1\right\rangle_2) + \left|01\right\rangle_{01}(\beta \left|0\right\rangle_2 + \alpha \left|1\right\rangle_2) + \left|10\right\rangle_{01}(\alpha \left|0\right\rangle_2 - \beta \left|1\right\rangle_2) + \left|11\right\rangle_{01}(\alpha \left|1\right\rangle_2 - \beta \left|0\right\rangle_2)\right] \\
=&\frac{1}{2}\left[\left|00\right\rangle_{01}\left|\psi\right\rangle_2 + \left|01\right\rangle_{01} X\left|\psi\right\rangle_2 + \left|10\right\rangle_{01} Z\left|\psi\right\rangle_2 + \left|11\right\rangle_{01} XZ\left|\psi\right\rangle_2\right]
\end{aligned}$$

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:

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

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

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

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 $\{e_1, \ldots, e_n\}$ 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 $\{e_1, \ldots, e_n\}$ 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 $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$. Now, we observe that for all $i \in [n]$

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

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**. In particular, this identity directly implies the following useful property. 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 $\overline{iu_2} = \bar{i} \cdot \overline{u_2} = -i\overline{u_2}$ 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

$$A^{\dagger}_{ij} = \overline{A_{ji}}$$

This property is incredibly useful, because it implies that the adjoint operator of A is its transposed conjugate matrix. Most notably, due to the way we defined our scalar product, it holds that for any column vector $|x\rangle$ we have that

$$\langle x| = |x\rangle^{\dagger}$$

which gives an intuition of the reason why we defined our scalar product as such.

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.

Proposition 2.9

If A and B are two unitary operators, then AB is a unitary operator.

Proof. Since A and B are unitary it holds that

$$A^\dagger A = AA^\dagger = B^\dagger B = BB^\dagger = I$$

Now, by Proposition 2.5 we have that

$$(AB)^\dagger = B^\dagger A^\dagger$$

from which we conclude that

$$(AB)^\dagger(AB) = B^\dagger A^\dagger AB = B^\dagger IB = B^\dagger B = I$$

□

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.

Lastly, say that we know that how an operator U acts on some input $|x\rangle$, and we want to derive U. How can we do this?

Proposition 2.10

For any operator U in a Hilbert space $\mathcal{H}$, if $\mathcal{B}$ is a base of $\mathcal{H}$ it holds that

$$U = \sum_{b\in\mathcal{B}} U\,|b\rangle\,\langle b|$$

Proof. The formula derives directly from the resolution of the identity, and the linearity of operators of Hilbert spaces

$$\begin{aligned} U &= U \cdot I \\ &= U \cdot \sum_{b \in \mathcal{B}} |b\rangle \langle b| \\ &= \sum_{b \in \mathcal{B}} U |b\rangle \langle b| \end{aligned}$$

□

Therefore, if we know how U acts component-wise, we can reconstruct the operator acting on the whole space as such.

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 2.11

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 surprisingly 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 $\{f_1, \ldots, f_n\}$ 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 $\{f_1, \ldots, f_n\}$ 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 2.12

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

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

□

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 2.13

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
   $$i\hbar \frac{\mathrm{d}}{\mathrm{d}t} v(t) = Hv(t)$$
   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):

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

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 intro

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 □ 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: $\quad q_1 = X(q_1)$
3: $\quad q_0, q_1 = (H \otimes H)(q_0, q_1)$
4: $\quad q_0, q_1 = U_f(q_0, q_1)$
5: $\quad q_0 = H(q_0)$
6: $\quad$ **return** measure(q_0)
7: **end function**

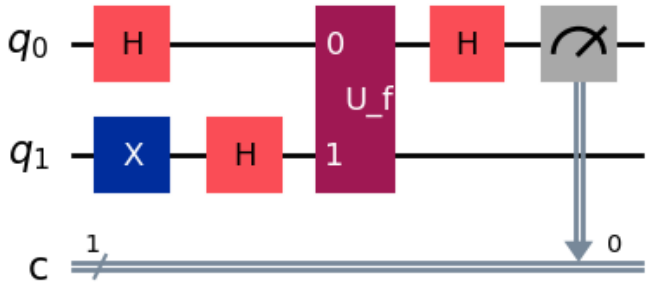


Figure 3.1: The quantum circuit for Deutsch's algorithm. The box colored in *magenta* labeled with `U_f` represents a "black-box" for whatever computation U_f represents (which directly depends on the chioce of f).

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 $a \in \{0,1\}^n$, 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 usefult 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 $q_0 = |0\rangle^{\otimes n}$ (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 $|0\rangle^{\otimes n}$ if f is constant, $|1\rangle$ otherwise.

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

Note that in this algorithm q_0 are actually n qubits, thus q_0 is initially set to $|0\rangle^{\otimes n}$. 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 drawing

For our discussion we will call $\mathbb{B} = \{0, 1\}$.

Claim 1: $\forall a \in \mathbb{B} \quad H|a\rangle = \frac{1}{\sqrt{2}} \sum_{b\in\mathbb{B}} (-1)^{a\cdot b} |b\rangle$.

Proof of the Claim. We observe that

$$H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = \frac{1}{\sqrt{2}} \sum_{b\in\mathbb{B}} (-1)^{0\cdot b} |b\rangle$$

and analogously

$$H|1\rangle = |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{1}{\sqrt{2}} \sum_{b\in\mathbb{B}} (-1)^{1\cdot b} |b\rangle$$

□

In the following claim, we will denote with the $\cdot$ symbol the "canonical" scalar product, i.e.

$$\forall x, y \in \mathbb{B}^n \quad x \cdot y := \sum_{i=1}^{n} x_i y_i$$

Claim 2: $\forall x \in \mathbb{B}^n \quad H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{a\in\mathbb{B}^n} (-1)^{x\cdot a} |a\rangle$

Proof of the Claim. By the previous claim, we have that

$$\begin{aligned} H^{\otimes n} |x\rangle &= \bigotimes_{i=1}^{n} H |x_i\rangle \\ &= \bigotimes_{i=1}^{n} \left(\frac{1}{\sqrt{2}} \sum_{b \in \mathbb{B}} (-1)^{x_i b} |b\rangle \right) |x_i\rangle \quad \text{(by Claim 1)} \\ &= \frac{1}{\sqrt{2}^n} \bigotimes_{i=1}^{n} (|0\rangle + (-1)^{x_i} |1\rangle) \\ &= \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{x \cdot a} |a\rangle \end{aligned}$$

□

Finally, we are ready to prove the correctness of the algorithm.

$$
\begin{aligned}
& q_0 \otimes q_1 \\
=& \left|0\right\rangle^{\otimes n} \otimes \left|0\right\rangle \\
\xrightarrow{X(q_1)} & \left|0\right\rangle^{\otimes n} \otimes \left|1\right\rangle \\
\xrightarrow{H^{\otimes n}(q_0,q_1)} & H^{\otimes n} \left|0\right\rangle^{\otimes n} \otimes H \left|1\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{0^n \cdot a} \left|a\right\rangle \otimes \left|-\right\rangle && \text{(by Claim 2)} \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left|a\right\rangle \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left(\left|a\right\rangle \otimes \left|-\right\rangle\right) \\
\xrightarrow{U_f(q_0,q_1)} & \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left((-1)^{f(a)} \left|a\right\rangle \otimes \left|-\right\rangle\right) && \text{(by Lemma 3.1)} \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left|a\right\rangle \otimes \left|-\right\rangle \\
\xrightarrow{H^{\otimes n}(q_0)} & H^{\otimes n} \left(\frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left|a\right\rangle\right) \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} \left((-1)^{f(a)} H^{\otimes n} \left|a\right\rangle\right) \otimes \left|-\right\rangle \\
=& \frac{1}{\sqrt{2}^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} \left(\frac{1}{\sqrt{2}^n} \sum_{b \in \mathbb{B}^n} (-1)^{a \cdot b} \left|b\right\rangle\right) \otimes \left|-\right\rangle && \text{(by Claim 2)} \\
=& \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} \sum_{b \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b} \left|b\right\rangle \otimes \left|-\right\rangle \\
=& \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} \sum_{b \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b} \left|b\right\rangle \otimes \left|-\right\rangle \\
=& \sum_{b \in \mathbb{B}^n} \left(\frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b}\right) \left|b\right\rangle_0 \otimes \left|-\right\rangle_1
\end{aligned}
$$

Now note that this state describes the superposition of the system, but the next step of the algorithm will only measure q_0, therefore we can ignore $\left|-\right\rangle_1$ and just focus on the amplitudes of q_0. Then, by calling

$$\forall b \in \mathbb{B}^n \quad \alpha_b := \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot b}$$

we can rewrite q_0 as follows

$$q_0 = \sum_{b \in \mathbb{B}^n} \alpha_b \left|b\right\rangle$$

Finally, since we want to determine the probability that q_0 collapses into the state $\left|0\right\rangle^{\otimes n}$

specifically, we can easily evaluate the associated amplitude of the latter, i.e.

$$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)+a \cdot 0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)}$$

From this, we can easily conclude that:

- if f is constant, then

  $$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}^n} (-1)^{f(a)} = \frac{1}{2^n} \cdot 2^n \cdot (-1)^b = (-1)^b$$

  where $b \in \mathbb{B}$, meaning that it is guaranteed that q_0 will collapse to 0^n

- if f is balanced, then

  $$\alpha_{0^n} = \frac{1}{2^n} \sum_{a \in \mathbb{B}} (-1)^{f(a)} = \frac{1}{2^n} \cdot 0 = 0$$

  meaning that it is guaranteed that q_0 will *not* collapse to 0^n

3.3 Grover's algorithm

Even though Deutsch-Josza algorithm is able to solve the decision problem we described through *one* single evaluation of f, it is fairly apparent that the problem their algorithm solves is quite artificial. However the next algorithm that we are going to discuss solves a problem that is definitely more useful.

In 1997, Grover [Gro97] published a landmark paper called "Quantum Mechanics Helps in Searching for a Needle in a Haystack", and the name already suggests the problem his work tried to solve: the search problem. The setting is the following: we are given an array of N elements — we can assume that N is always a power of 2 for some n, i.e. $N = 2^n$ — that contains M "solution" elements. However, we do not know their positions, and the problem asks to find the index of any solution element.

More formally, given a Boolean function

$$f : \{0, \ldots, N-1\} \to \mathbb{B} : x \mapsto \begin{cases} 1 & A[x] \in S \\ 0 & \text{otherwise} \end{cases}$$

where A is our array, and S is the set of solution elements, the problem asks to return an x such that $f(x) = 1$, i.e. such that $A[x]$ is a solution.

With a classical computation, it is easy to see that we need $O\left(\frac{N}{M}\right)$ accesses to A to solve our problem, however we will see that the algorithm Grover developed is able to return a "solution index" in $O\left(\sqrt{\frac{M}{N}}\right)$ with *high probability*, i.e. Grover's algorithm provides a **quadratic** speedup compared to any classical algorithm — however, it is probabilistic.

Before explaining the details of the algorithm, we need to define some new operators that will be used in Grover's algorithm, and introduce some general notation:

- given an arbitrary qubit $|\psi\rangle$, we will write its superposition of states as follows

$$|\psi\rangle = \sum_{b \in \mathbb{B}^n} \alpha_b |b\rangle$$

  where $\sum_{b \in \mathbb{B}^n} |\alpha_b| = 1$

- we define an operator O_f (where f is the indicator function of the array defined before) that computes as follows:

$$\forall x \in \mathbb{B}^n \quad O_f |x\rangle := (-1)^{f(x)} |x\rangle$$

- given an arbitrary qubit $|\psi\rangle$, we define a new operator W as follows:

$$W := 2 |s\rangle \langle s| - I$$

  where $|s\rangle$ is the **uniform superposition**, a superposition of states in which each amplitude is equally likely

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}^n} |x\rangle$$

- finally, we will define an operator G that will simply compose the last two operators we described

$$G = W \cdot O_f$$

Algorithm 3.3: Grover's algorithm

Given a Boolean function f that describes the solution elements of an array having M solution elements, and n qubits, the algorithm returns a solution index with high probability TODO SPIEGA CHE VUOL DIRE.

1: **function** GROVER(f, q_0)
2: $q_0 = H^{\otimes n}(q_0)$
3: **for** $i \in \left[1, O\left(\sqrt{\frac{N}{M}}\right)\right]$ **do** ▷ where $N = 2^n$
4: $q_0 = G(q_0)$
5: **end for**
6: **return** measure(q_0)
7: **end function**

First of all, we see that this algorithm takes only n qubits as input, however the actual implementation of the algorithm is slightly different, as shown in the following quantum circuit:

TODO drawing

Indeed, we can see that the real quantum circuit takes $n + 1$ inputs, and the additional register is called "ancilla" becuase its only purpose is to actually implement the O_f operator, which is designed exactly as if it was the U_f black-box we discussed in previous

algorithms. This can be done thanks to Lemma 3.1, which guarantees that if we give $|-\rangle$ as the second input to U_f we get

$$O_f |x\rangle \otimes |-\rangle = (-1)^{f(x)} \otimes |-\rangle$$

which implies that the ancilla register will sill be $|-\rangle$, therefore we can just ignore the second register completely throughout the whole computation, and we are sure that O_f computes correctly.

Furthermore, before proceeding, let us prove that G is actually a unitary operator, i.e. that this is a valid quantum computation.

Claim: The operator G is unitary.

Proof of the Claim. By Proposition 2.9 we know that proving that W and O_f are unitary is sufficient to prove the claim. □

todo da finire

Now that we know this operator is unitary, we can delve into the details of the algorithm. To see what happens at each iteration, we will describe the complete state of the system in a rather unusual way. Let $|a\rangle$ and $|b\rangle$ be the following superpositions:

$$|a\rangle := \frac{1}{\sqrt{N-M}} \sum_{x \in \bar{S}} |x\rangle \qquad |b\rangle := \frac{1}{\sqrt{M}} \sum_{x \in S} |x\rangle$$

In other words, $|a\rangle$ is the uniform superposition of non-solution indices, and $|b\rangle$ is the superposition of solution ones. To explain the $\frac{1}{\sqrt{M}}$ factor in front of $|b\rangle$, we recall that in quantum mechanics all state vectors must be normalized, and since the squared norm of $|\tilde{b}\rangle$ (which will denote $|b\rangle$ without the multiplicative factor in front) is equal to

$$\begin{aligned}
\langle \tilde{b} | \tilde{b} \rangle &= \left(\sum_{x \in S} |x\rangle \right)^{\dagger} \left(\sum_{x \in S} |x\rangle \right) \\
&= \left(\sum_{x \in S} \langle x| \right) \left(\sum_{y \in S} |y\rangle \right) \\
&= \sum_{x \in S} \sum_{y \in S} \langle x | y \rangle \\
&= \sum_{x \in S} \sum_{y \in S} \delta_{xy} \qquad \text{(basis states are orthonormal)} \\
&= \sum_{x \in S} 1 \\
&= |S| \\
&= M
\end{aligned}$$

the norm of $|\tilde{b}\rangle$ is $\sqrt{M}$, hence to normalize it it suffices to add $\frac{1}{\sqrt{M}}$ in front of $|\tilde{b}\rangle$— the reasoning for $|a\rangle$ is analogous. Moreover, it's easy to see that $|a\rangle$ and $|b\rangle$ are orthogonal, so they form an orthonormal basis for a 2D space.

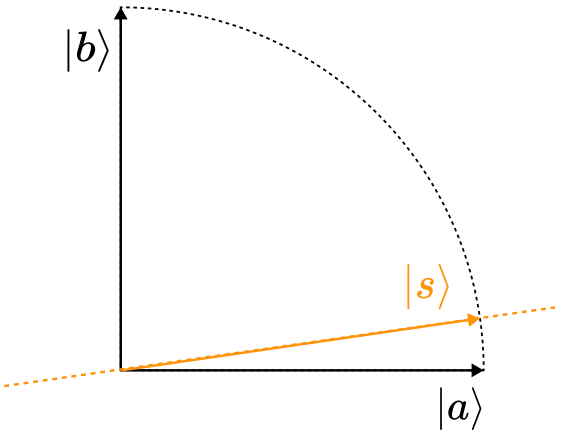
Now, because of how we defined $|a\rangle$ and $|b\rangle$, we can rewrite the uniform superposition

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}} |x\rangle$$

as shown below

$$|s\rangle = \sqrt{\frac{N-M}{N}} |a\rangle + \sqrt{\frac{M}{N}} |b\rangle$$

This is quite interesting, because it means that we can describe $|s\rangle$ in terms of $|a\rangle$ and $|b\rangle$. Let us do exactly this, and plot the resulting graph on a 2D space that has $|a\rangle$ and $|b\rangle$ as axis.



We observe that:

- both $|s\rangle$, $|a\rangle$ and $|b\rangle$ are normalized, and all the vectors that we are going to consider are quantum states, so they will be normalized too, therefore we can restrict our focus on a 2D circumference of radius 1 — this plane is usually called **Grover plane**
- since we expect that $M \ll N$, we have that $\sqrt{\frac{M}{N}} \ll \sqrt{\frac{N-M}{N}}$, which basically means that the vector $|s\rangle$ is almost parallel to $|a\rangle$ (the bigger is the numer of solution indices, the bigger the angle between $|s\rangle$ and $|a\rangle$)

Consider any state $|\psi\rangle = \sum_{x \in \mathbb{B}^n} \alpha_x |x\rangle$; we observe that

$$\begin{aligned} O_f |\psi\rangle &= O_f \sum_{x \in \mathbb{B}^n} \alpha_x |x\rangle \\ &= \sum_{x \in \mathbb{B}^n} \alpha_x O_f |x\rangle \\ &= \sum_{x \in \mathbb{B}^n} \alpha_x (-1)^{f(x)} |x\rangle \end{aligned}$$

which basically means that each time we apply the O_f operator we are flipping the sign of the amplitudes of the compoents of $|\psi\rangle$ that represent solution indices. In other words, the O_f operator flips its input w.r.t. $|a\rangle$.

Moreover, we observe that any state $|\psi\rangle$ can be decomposed into

$$|\psi\rangle = \alpha\,|s\rangle + |\psi_\perp\rangle$$

where $\alpha\,|s\rangle$ is the projection of $|\psi\rangle$ along $|s\rangle$'s space — thus $\alpha = \langle s|\psi\rangle$ — and $|\psi_\perp\rangle$ is the projection of $|\psi\rangle$ along the space that is orthogonal to $|s\rangle$'s. Therefore, we have that

$$\begin{aligned}
W\,|\psi\rangle &= W(\alpha\,|s\rangle + |\psi_\perp\rangle) \\
&= 2\,|s\rangle\,\langle s|\,(\alpha\,|s\rangle + |\psi_\perp\rangle) - (\alpha\,|s\rangle + |\psi_\perp\rangle) \\
&= 2\alpha\,|s\rangle\,\langle s|s\rangle + 2\,|s\rangle\,\langle s|\psi_\perp\rangle - (\alpha\,|s\rangle + |\psi_\perp\rangle) \\
&= 2\alpha\,|s\rangle \cdot 1 + 0 - (\alpha\,|s\rangle + |\psi_\perp\rangle) \\
&= \alpha\,|s\rangle - |\psi_\perp\rangle
\end{aligned}$$

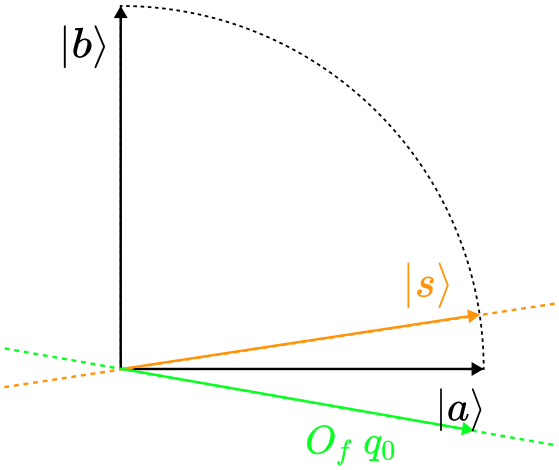
which means that what W actually performs is leaving the component along $|s\rangle$'s space unchanged, and it flips the sign of the component of the orthogonal space. In other words, what W computes is the reflection of $|\psi\rangle$ w.r.t. $|s\rangle$.

We can finally describe Grover's algorithm in detail. First, we see that

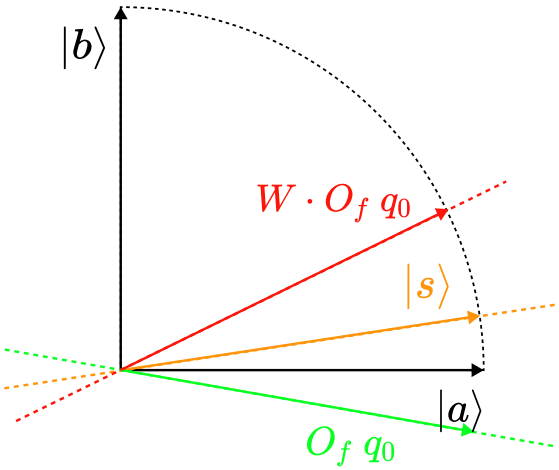
$$\begin{aligned}
& q_0 \\
&= |0\rangle^{\otimes n} \\
&\xrightarrow{H^{\otimes n}(q_0)} \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} (-1)^{0^n \cdot x}\,|x\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}^n} |x\rangle \\
&= |s\rangle
\end{aligned}$$

Indeed, the only purpose of the first Hadamard operator is to "move" the initial state slightly away from $|a\rangle$ — and also making each component initially equally likelly. Now, let's see what happens at each application of the $G = W \cdot O_f$ operator:

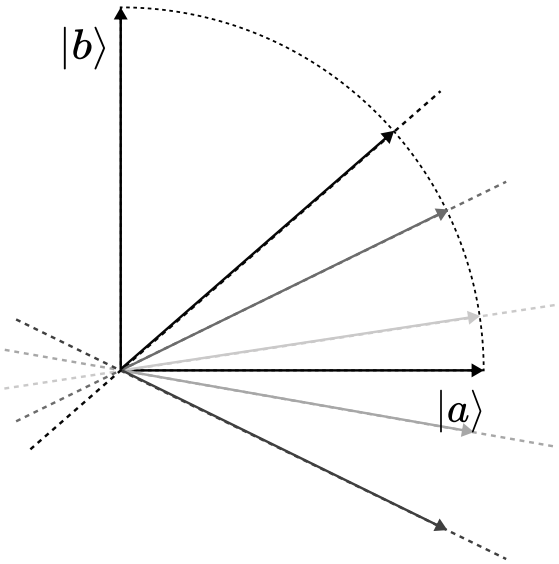
- as previously shown, the operator O_f reflects its input w.r.t. the $|a\rangle$ axis — below we show what happens when we first apply $O_f\ q_0 = O_f\,|s\rangle$:



- additionally, as previously explained the operator W reflects its input w.r.t. the axis described by the $|s\rangle$ vector, thus when we compute $W \cdot O_f\ q_0$ we will end up with the following vector:



- this suggests that each time we apply the operator G we are making q_0 closer and closer to $|b\rangle$, as depicted below:



This is the core idea of Grover's algorithm: if we rewrite q_0 in terms of $|a\rangle$ and $|b\rangle$

$$q_0 = \beta_0 |a\rangle + \beta_1 |b\rangle$$

we get that through Grover's algorithm we transformed q_0 such that it is now very close to $|b\rangle$, meaning that $\beta_0 \ll \beta_1$. This direcly implies that when we will measure q_0 at the end of Grover's procedure the likelihood that it will collapse into some $|x\rangle$ that is a component of $|b\rangle$ — i.e. a solution index — is very high. In other words, what happens with Grover's algorithm is that we gradually increase the amplitudes of the solution indices, in order to maximize the probability that our qubit will collapse in one them when it will be measured at the end of the procedure.

Now that we know how Grover's algorithm works, the only thing left to discuss is the $O\left(\sqrt{\frac{N}{M}}\right)$. Why is it guaranteed that after this amount of applications of the G operator we are done with the algorithm? Well, we actually have the opposite problem: in reality, we have to *stop early enough*. Consider again how Grover's algorithm operates in the Grover plane; clearly, if we apply G too many times, what happens is that q_0 will end up past $|b\rangle$ itself: drawing

Let the angle between $|a\rangle$ and $|s\rangle$ be θ; since O_f flips q_0 w.r.t. $|a\rangle$, and W flips O_f q_0 w.r.t. $|s\rangle$, G will cumulatively rotate q_0 by 2θ: Indeed, with each application of G we are drawing rotating q_0 by 2θ, which means that at the k-th application it holds that

$$G^k \; q_0 = \cos(2k+1)\theta \, |a\rangle + \sin(2k+1)\theta \, |b\rangle$$

for any $k \in \mathbb{N}$, where the additional 1 comes from the fact that $q_0 = |s\rangle$ through the Hadamard transformation at the start of the process. Thus, to evaluate the optimal number of iterations we need to find the optimal k, i.e. the one that maximizes the probability of measuring a solution index, which is equal to the squared amplitude of $|b\rangle$, namely

$$\Pr[\text{measure}(q_0) = |b\rangle] = \sin^2(2k+1)\theta$$

Hence, we have that $\sin^2(2k+1)\theta = 1$ when $(2k+1)\theta = \frac{\pi}{2}$, and solving for k we get that

$$k = \frac{\pi}{4\theta} - \frac{1}{2}$$

Lastly, since θ is the angle between $|a\rangle$ and $|s\rangle$, we can rewrite $|s\rangle$ as

$$|s\rangle = \cos\theta\,|a\rangle + \sin\theta\,|b\rangle$$

which directly implies that

$$\sin\theta = \sqrt{\frac{M}{N}} \iff \theta = \arcsin\sqrt{\frac{M}{N}}$$

and therefore

$$\begin{aligned} k &= \frac{\pi}{4\theta} - \frac{1}{2} \\ &= \frac{\pi}{4\arcsin\sqrt{\frac{M}{N}}} - \frac{1}{2} \\ &\leq \frac{\pi}{4\arcsin\sqrt{\frac{M}{N}}} \\ &\leq \frac{\pi}{4\sqrt{\frac{M}{N}}} \\ &= \frac{\pi}{4}\sqrt{\frac{N}{M}} \\ &= O\left(\sqrt{\frac{N}{M}}\right) \end{aligned}$$

which finally explains the quadratic speedup of Grover's algorithm w.r.t. the classical version of the problem.

As a final note, we observe that Grover's algorithm assumes that $\theta \leq \frac{\pi}{4}$, otherwise we overshoot $|b\rangle$ with a single iteration of the G operator — since $\theta > \frac{\pi}{4}$ would mean that $|s\rangle$ is placed on the "upper half" of the Grover plane. However, to ensure this constraint on θ we only need that $M \leq \frac{N}{2}$, i.e. at most half of the elements in our array are solution elements. Indeed, we see that

$$M \leq \frac{N}{2} \implies \sin\theta = \sqrt{\frac{M}{N}} \leq \sqrt{\frac{1}{2}} \implies \theta \leq \arcsin\sqrt{\frac{1}{2}} = \frac{\pi}{4}$$

What can we do if the number of solutions is more than half the size of the array? We simply invert the problem and find the non-solutions!

3.3.1 Another perspective

Interestingly enough, we can look at what happens to the amplitudes of q_0 from another perspective. Usually, the W operator is called **diffusion operator**, and what it does is computing the so called *inversion about the mean* of its input. Consider any state $|\psi\rangle = \sum_{x\in\mathbb{B}^n} \alpha_x |x\rangle$, and observe that

$$\begin{aligned}
\langle s|\psi\rangle &= \left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}|y\rangle\right)^\dagger \left(\sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\right) && \text{(since } \langle s| = |s\rangle^\dagger\text{)}\\
&= \left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\langle y|\right)\left(\sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\right)\\
&= \frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\sum_{x\in\mathbb{B}^n}\alpha_x\langle y|x\rangle\\
&= \frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}\sum_{x\in\mathbb{B}^n}\alpha_x\delta_{xy} && \text{(basis states are orthonormal)}\\
&= \frac{1}{\sqrt{N}}\sum_{x\in\mathbb{B}^n}\alpha_x\sum_{y\in\mathbb{B}^n}\delta_{xy}\\
&= \frac{1}{\sqrt{N}}\sum_{x\in\mathbb{B}^n}\alpha_x\\
&= \sqrt{N}\overline{\alpha_\psi}
\end{aligned}$$

where $\overline{\alpha_\psi}$ is the average amplitude of $|\psi\rangle$. This implies that

$$\begin{aligned}
W|\psi\rangle &= (2|s\rangle\langle s| - I)|\psi\rangle\\
&= 2|s\rangle\langle s|\psi\rangle - \sum_{x\in\mathbb{B}^n}|x\rangle\\
&= 2|s\rangle\left(\sqrt{N}\overline{\alpha_\psi}\right) - \sum_{x\in\alpha_x|x\rangle} && \text{(for the previous observation)}\\
&= 2\left(\frac{1}{\sqrt{N}}\sum_{y\in\mathbb{B}^n}|y\rangle\right)\sqrt{N}\overline{\alpha_\psi} - \sum_{x\in\mathbb{B}^n}\alpha_x|x\rangle\\
&= \sum_{x\in\mathbb{B}^n}(2\overline{\alpha_\psi} - \alpha_x)|x\rangle
\end{aligned}$$

In other words, when we apply W to a superposition of states, what happens is that each amplitude of the basis states is trasformed as follows:

$$W : \alpha_x \to 2\overline{\alpha_\psi} - \alpha_x$$

To understand why this is important, let's look at what happens in Grover's algorithm after the first Hadamard application. TODO da finire

3.3.2 Fixed-Point Quantum Search

As we saw in the previous sections, with Grover's algorithm we need to be careful on how many times we apply the G operator, however we observe that the right value for k — i.e.

the numeber of iterations — strictly depends on both N and M. Assuming that N, the length of the array, is known, it is not guaranteed that we know M too (the number of solutions in the array). Therefore, if we want to apply Grover's algorithm we also need to be able to to a rough estimate on M, otherwise we might end up stopping either too early or too late. This makes Grover's algorithm fragile in some real-world scenarios.

Grover was actually aware of this problem, and in 2005 he designed a new algorithm which is able to solve this issue [Gro05]. The idea of the algorithm is to *monotonically* increase the probability of finding a solution, without oscillating back down $|a\rangle$. This algorithm is called **Fixed-Point Quantum Search**, becuase the solutions actually become a "stable fixed point" of the transformation — i.e. once you reach a good solution, further iterations leave it basically unchanged. Indeed, with Grover's search each iteration is a constant-angle rotation, while in fixed-point search we will see that the phase angles in each rotation changes such that the rotation angle decreases over time.

First, we need to define two new operators:

$$R_s := I - (1 - e^{i\theta}) |s\rangle \langle s| \qquad R_t := I - (1 - e^{i\theta}) |t\rangle \langle t|$$

where $|s\rangle$ is the starting state and $|t\rangle$ is the target state (in Grover's algorithm this was $|b\rangle$) — this is the original notation that Grover used in his paper, and actually explains why we used $|s\rangle$ in the previous version of the algorithm, it's just the "start". These two operators are called **phase shift operators**; as usual, before proceeding we need to show that both operators are actually unitary.

Claim: R_t and R_s are unitary operators.

Proof of the Claim. TODO □ todo

What are these two operators in the first place? When we presented the W operator, we also noticed how it actually performes a reflection of its input w.r.t. the space of $|s\rangle$. Well, through a very similar argument it can be shown that

$$W^{\perp} := I - 2 |s\rangle \langle s|$$

performs a reflection of its input w.r.t. the space *orthogonal* to $|s\rangle$ — indeed, we end up with

$$W^{\perp} |\psi\rangle = |\psi_{\perp}\rangle - \alpha |s\rangle$$

If we now look at the R_s operator, we can see that when we actually compute the reflection it performs we end up with

$$R_s |\psi\rangle = |\psi_{\perp}\rangle + e^{i\theta}\alpha |s\rangle$$

This suggests that what R_s actually computes is a "soft reflection" w.r.t. the space perpendicular to $|s\rangle$. Through an analogous argument, we can see that

$$R_t |\psi\rangle = |\psi_{\perp}\rangle + e^{i\theta}\alpha |t\rangle$$

meaning that R_t computes a "soft reflection" w.r.t. the space perpendicular to $|t\rangle$ — however, we observe that the latter is literally the space of $|a\rangle$, indeed O in Grover's algorithm could have been defined as

$$O = I - 2 |b\rangle \langle b|$$

and indeed it is sometimes defined such.

Now, we are going to define an addition operator called U as such: let U be any unitary operator such that, for some small $\varepsilon > 0$, it holds that

$$|\langle t|Us\rangle|^2 = 1 - \varepsilon$$

We observe that

- by the laws of quantum mechanics we have that

  $$\Pr[\text{measure}\,(U\,|s\rangle) = |t\rangle] = |\langle t|Us\rangle|^2$$

  therefore we require U to be an operator such that it "drives $|s\rangle$ close to $|t\rangle$ with high probability" — i.e. $1 - \varepsilon$
- we know that U exists since it's just a rotation in a 2D space
- also, a geometric interpretation of the scalar product is that we are measuring the cosine of the angle between $|t\rangle$ and $|Us\rangle$, and we want this value to be very high (such that the angle woule be very small) — we obeserve that we are in Hilbert spaces so the notion of "angle" is not the same of the one we are used to with Euclidean spaces, but this is just to have an idea of what is happening with U

Moreover, define the following operator

$$G := UR_sU^\dagger R_tU$$

Let's see what happens when we evaluate $G\,|s\rangle$. By denoting $U_{ts} = \langle t|Us\rangle$, we can prove the following equality.

Lemma 3.2

It holds that

$$G\,|s\rangle = U\,|s\rangle\left[e^{i\theta} + |U_{ts}|^2\left(e^{i\theta} - 1\right)^2\right] + |t\rangle\,U_{ts}\left(e^{i\theta} - 1\right)$$

Proof. First, we prove the following equality.

Claim: It holds that $\langle s|\,U^\dagger\,|t\rangle\,U_{ts} = |U_{ts}|^2$.

Proof of the Claim.

$$\begin{aligned}
\langle s|\,U^\dagger\,|t\rangle\,U_{ts} &= \langle s|U^\dagger t\rangle\,U_{ts} \\
&= \langle Us|t\rangle\,U_{ts} && \text{(by def. of adjoint)} \\
&= \overline{\langle t|Us\rangle}U_{ts} && \text{(by prop. of scalar products)} \\
&= \overline{U_{ts}}U_{ts} \\
&= |U_{ts}|^2 && (z\cdot\overline{z} = |z|^2)
\end{aligned}$$

□

Let $P_s = |s\rangle\langle s|$ and $P_t = |t\rangle\langle t|$; thus, we have that

$$\begin{aligned}
G &= UR_sU^\dagger R_tU \\
&= U(I-(1-e^{i\theta})P_s)U^\dagger(I-(1-e^{i\theta})P_t)U \\
&= (U-(1-e^{i\theta})UP_s)U^\dagger(U-(1-e^{i\theta})UP_t) \\
&= (UU^\dagger-(1-e^{i\theta})UP_sU^\dagger)(U-(1-e^{i\theta})UP_t) \\
&= (I-(1-e^{i\theta})UP_sU^\dagger)(U-(1-e^{i\theta})UP_t) && (UU^\dagger = I) \\
&= U-(1-e^{i\theta})P_tU-(1-e^{i\theta})UP_s+(1-e^{i\theta})^2UP_sU^\dagger P_tU
\end{aligned}$$

Therefore, we find that $G|s\rangle$ can be evaluated as follows:

$$\begin{aligned}
G|s\rangle &= \left(U-(1-e^{i\theta})P_tU-(1-e^{i\theta})UP_s+(1-e^{i\theta})UP_sU^\dagger P_tU\right)|s\rangle \\
&= U|s\rangle-(1-e^{i\theta})P_tU|s\rangle-(1-e^{i\theta})UP_s|s\rangle+(1-e^{i\theta})^2UP_sU^\dagger P_tU|s\rangle \\
&= U|s\rangle-(1-e^{i\theta})|t\rangle U_{ts}-(1-e^{i\theta})U|s\rangle+(1-e^{i\theta})^2U|s\rangle\langle s|U^\dagger|t\rangle U_{ts} \\
&= U|s\rangle-(1-e^{i\theta})|t\rangle U_{ts}-(1-e^{i\theta})U|s\rangle+(1-e^{i\theta})^2U|s\rangle|U_{ts}|^2 && \text{(by the claim)} \\
&= \dots && \text{(algebraic manipulation)} \\
&= U|s\rangle\left[e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right]+|t\rangle U_{ts}\left(e^{i\theta}-1\right)
\end{aligned}$$

□

Most importantly, this equality can be used in the following proposition, which shows that we can actually find an angle θ for which the distance from $|t\rangle$ decreases significantly.

Proposition 3.2

There exists an angle θ such that

$$\Pr[\text{measure}(G\ |s\rangle) = |t\rangle] = 1-\varepsilon^3$$

Proof. Thanks to the previous lemma, we obtain that

$$\begin{aligned}
\Pr[\text{measure}\,(G|s\rangle) = |t\rangle] &= |\langle t|Gs\rangle|^2 \\
&= \left|\langle t|\left[U|s\rangle\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+|t\rangle U_{ts}\left(e^{i\theta}-1\right)\right]\right|^2 \\
&= \left|\langle t|Us\rangle\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+\langle t|t\rangle U_{ts}\left(e^{i\theta}-1\right)\right|^2 \\
&= \left|U_{ts}\left(e^{i\theta}+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right)+U_{ts}\left(e^{i\theta}-1\right)\right|^2 \\
&= \left|U_{ts}\left[2e^{i\theta}-1+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right]\right|^2 \\
&= |U_{ts}|^2\cdot\left|2e^{i\theta}-1+|U_{ts}|^2\left(e^{i\theta}-1\right)^2\right|^2 \\
&= (1-\varepsilon)\cdot\left|2e^{i\theta}-1+(1-\varepsilon)\left(e^{i\theta}-1\right)^2\right|^2 && \left(|U_{ts}|^2 = 1-\varepsilon\right)
\end{aligned}$$

Now, let's see what happens if we set $\theta = \frac{\pi}{3}$: we obtain that

$$\begin{aligned}
\Pr[\text{measure}\,(G\,|s\rangle) = |t\rangle] &= (1-\varepsilon)\cdot\left|2e^{i\frac{\pi}{3}} - 1 + (1-\varepsilon)\left(e^{i\frac{\pi}{3}} - 1\right)^2\right|^2 \\
&= (1-\varepsilon)\cdot\left|2e^{i\frac{\pi}{3}} - 1 - (1-\varepsilon)i\tfrac{\pi}{3}\right|^2 && \left(\left(e^{i\frac{\pi}{3}} - 1\right)^2 = -e^{i\frac{\pi}{3}}\right) \\
&= (1-\varepsilon)\cdot\left|(1+\varepsilon)e^{i\frac{\pi}{3}} - 1\right|^2 \\
&= (1-\varepsilon)\cdot\left|(1+\varepsilon)\left(\frac{1}{2} + i\frac{\sqrt{4}}{2}\right) - 1\right|^2 && \text{(by Euler's formula)} \\
&= (1-\varepsilon)\cdot\left|\frac{1}{2} + i\frac{\sqrt{3}}{2} + \frac{\varepsilon}{2} + i\frac{\sqrt{3}}{2}\varepsilon - 1\right|^2 \\
&= (1-\varepsilon)\cdot\left|\frac{\varepsilon}{2} - \frac{1}{2} + i\frac{\sqrt{3}}{2}(1+\varepsilon)\right|^2 \\
&= (1-\varepsilon)\cdot\left[\left(\frac{\varepsilon}{2} - \frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}(1+\varepsilon)\right)^2\right] && \left(|z|^2 = \Re^2(z) + \Im^2(z)\right) \\
&= (1-\varepsilon)\cdot(1+\varepsilon+\varepsilon^2) \\
&= 1+\varepsilon+\varepsilon^2-\varepsilon-\varepsilon^2-\varepsilon^3 \\
&= 1-\varepsilon^3
\end{aligned}$$

This shows that by applying G the probability of measuring $|t\rangle$ has increased from $1-\varepsilon$ to $1-\varepsilon^3$. □

Indeed, it can be shown that by defining the following recursive sequence of operators

$$\begin{cases} U_0 := U & m = 0 \\ U_m = U_{m-1}R_sU_{m-1}^\dagger R_t U_{m-1} & m \geq 1 \end{cases}$$

we get that

$$\Pr[\text{measure}(|U_m\rangle\,|s\rangle) = |t\rangle] = \langle\langle t|U_m s\rangle\rangle^2 = 1-\varepsilon^{2q_m+1}$$

where q_m is the number of queries of $f(x)$.

TODO

drawing of the vector that grows monotonically

Unfortunately, there already exists a classical probabilistic algorithm that the failure probability drops as ε^{q+1} after q queries to f. Thus, the quantum advantage with this method is lost.

So, what do we do now? In 2014 Yoder, Low, and Chuang [YLC14] proposed a fixed-point quantum serach algorithm that monotonically converges to the target state while still retaining the quadratic advantage of the original Grover's algorithm over classical algorithms. The algorithm involves phase-shift operators that are parametrized with angles different from $\theta = \frac{\pi}{3}$, and again involes building a sequence of operators using

said phase shifts. However, the details of this result are way beyond the scope of our discussion, so we won't describe the details of their findings.

4

Shor's algorithm

Shor's algorithm is a quantum algorithm for finding the *prime factors of an integer*, developed in 1994 by the American mathematician Shor [Sho]. It is one of the few known quantum algorithms with compelling potential applications and strong evidence of **superpolynomial** speedup compared to best known classical (non-quantum) algorithm. It is one of the most famous algorithms in the field, and it sparked a lot of interest in recent years due to the implications of its theoretical speedup.

Shor's algorithm utilizes a lot of mathematical tools

da finire

Before explaining Shor's algorithm, we first need to present a very useful tool that is widely employed in various areas of computer science and mathematics, such as integer multiplication, signal processing (e.g. speech recognition, audio compression) and much more.

4.1 A recap on the Discrete Fourier Transform

The **Discrete Fourier Transform (DFT)** is a very powerful tool used in many applications and probably most widely known for its use in signal processing. Indeed, the Fourier transform is used to transform a signal defined on the *time domain* into its equivalent in the *frequency domain*. We will briefly explore the most important concepts that define the DFT to have a better understanding in order to the progress with its quantum counterpart.

First, what is the "time domain"? When we sample a signal, we usually sample such signal through some finite amount of time, and we discretize such time interval — this is why we consider the Discrete Fourier transform. What we are interested in is to transform such signal into the "frequency domain", which essentially means to understand what frequencies of sinusoids contribute to our signal and in which percentage. To do this, we will use some concepts of Linear Algebra. Say that the signal we sampled is discretized into N parts, thus our sample lives in the $\mathbb{R}^N$ space — it is nothing but a vector of N complex values which describe the signal at each timestep. When we say "time domain",

what we mean is basically this exact vector, i.e. expressed in the canonical orthonormal basis

$$e_0 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad e_1 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad \dots \quad e_{N-1} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

(we will start counting at 0 for convenience sake). In other words, if our signal is defined by some vector

$$\mathbf{x} = \begin{pmatrix} t_0 \\ t_1 \\ \vdots \\ t_{N-1} \end{pmatrix}$$

it holds that

$$\mathbf{x} = \sum_{n=0}^{N-1} t_n e_n$$

Then, what is the "frequency domain"? We need some preliminary observations to describe it. Our goal is to take into account the frequencies that define our signal, therefore we need to move into a space in which frequencies are "first-class citizens". The idea is to use each possible sinusoid by varying the frequency, add up their contributions, and weight the latter in the precise way that allows us to reconstruct our original signal. Let us fix a component $n \in [0, N-1]$, and first consider only cosinusoids, for instance

$$\cos(2\pi \cdot 0 \cdot n) \quad \cos(2\pi \cdot 1 \cdot n) \quad \cos(2\pi \cdot 2 \cdot n) \quad \dots \quad \cos(2\pi \cdot (N-1) \cdot n)$$

Basically, we are trying to construct a basis built on each possible frequency f by obtaining a basis vector $\cos(2\pi \cdot f \cdot n)$. Moreover, we will consider $\cos(2\pi mn/N)$ for $m \in [0, N-1]$ since it can be shown that the number of possible frequencies that can be represented on a N-sized time window is exactly N, and to scale the frequency accordingly we just need to divide m over N, the size of the sample.

Is this enough to reach our goal? Can we describe any signal in this way? Well, we observe that m is ranging from 0 to $N-1$, but $\cos(-\theta) = \cos(\theta)$, which implies that

$$m > \frac{N}{2} \implies \cos(2\pi mn/N) = \cos(2\pi(N-m)n/N)$$

In other words, basically half of our basis vectors add no information at all. Indeed, the span of the vectors we chose has size

$$\dim\left(\text{span}\left(\{\cos(2\pi \cdot m \cdot n/N)\}_{m=1}^{N-1}\right)\right) = \begin{cases} \frac{N}{2}+1 & N \text{ is even} \\ \frac{N+1}{2} & N \text{ is odd} \end{cases}$$

where the last added 1 comes from the fact that when $m = 0$ we generate $\cos(0) = 1$ which is really linearly independent from the others cosines. This suggests that cosinusoids alone are not enough to describe our space.

Hence, the most natural thing that we can do is add the contributions of sinusoids as well. A geometric interpretation of the fact that cosinusoids are not enough is that sinusoids are just phase-shifted cosines, but the shift in phase is not captured by changing the frequencies of the cosines alone. We need the contributions of some "altered" cosinusoids — in terms of phases — to get an actual basis and be able to represent any possible vector. Hence, let's consider additional N sinusoids

$$\sin(2\pi \cdot 0 \cdot n) \quad \sin(2\pi \cdot 1 \cdot n) \quad \sin(2\pi \cdot 2 \cdot n) \quad \dots \quad \sin(2\pi \cdot (N-1) \cdot n)$$

Do we get a base of size more than N then? We observe that $\sin(\theta) = -\sin(-\theta)$, therefore we have that

$$m > \frac{N}{2} \implies \sin(2\pi mn/N) = -\sin(2\pi(N-m)n/N)$$

which again it implies that half of these sinusoids add no useful information. However, in this case we also have the fact that

$$\sin(2\pi \cdot 0 \cdot n/M) = \sin(0) = 0$$

and for N even it also happens that

$$\sin(2\pi \cdot (N/2) \cdot n/N) = \sin(\pi n) = 0$$

which means that these two sinusoid cannot be considered because they are the 0 vector of this space. Therefore, we get that

$$\dim\left(\text{span}\left(\{\cos(2\pi \cdot m \cdot n/N)\}_{m=1}^{N-1}\right)\right) = \begin{cases} \frac{N}{2} - 1 & N \text{ is even} \\ \frac{N-1}{2} & N \text{ is odd} \end{cases}$$

Finally, this means that putting all these cosines and sinusoids together we form a base for a space that has size

$$\begin{cases} \frac{N}{2} + 1 + \frac{N}{2} - 1 = N & N \text{ is even} \\ \frac{N+1}{2} + \frac{N-1}{2} = N & N \text{ is odd} \end{cases} = N$$

Hence, we can fully describe $\mathbb{R}^N$, namely for any vector $\mathbf{x} \in \mathbb{R}^N$ we have that

$$\forall n \in [0, N-1] \quad \mathbf{x}(n) = \sum_{m=0}^{N-1} \alpha_m \cos(2\pi mn/N) + \sum_{m=0}^{N-1} \beta_m \sin(2\pi mn/N)$$

by describing the vector component-wise.

Furthermore, we observe that this basis is actually orthogonal. prove it, boring

Now, the last thing that we need to do is move into the complex space $\mathbb{C}^N$. Thankfully, we can leverage the very powerful Euler's formula

$$e^{i\theta} = \cos\theta + i\sin\theta$$

to immediately some the contributions of cosinusoids and sinusoids into a single value $e^{2\pi imn/N}$. In other words, for any vector $\mathbf{x} \in \mathbb{C}^N$ it holds that

$$\forall n \in [0, N-1] \quad \mathbf{x}(n) = \sum_{m=0}^{N-1} \gamma_m e^{2\pi imn/N}$$

The basis of this space is thus

$$e^{2\pi i \cdot 0 \cdot x/N} \quad e^{2\pi i \cdot 1 \cdot x/N} \quad \dots \quad e^{2\pi i \cdot (N-1) \cdot n/N}$$

We observe that this basis is orthogonal as well, but it's not orthonormal in fact Hence, because the norm of each vector is

prove it, boring too

$$\begin{aligned}
\sqrt{\langle e^{2\pi imn/N} | e^{2\pi imn/N} \rangle} &= \sqrt{\sum_{n=0}^{N-1} e^{2\pi imn/N} \cdot \overline{e^{2\pi imn/N}}} \\
&= \sqrt{\sum_{n=0}^{N-1} |e^{2\pi imn/N}|^2} \\
&= \sqrt{\sum_{n=0}^{N-1} 1} \\
&= \sqrt{N}
\end{aligned}$$

we usually consider the same base but scaled by a factor of $\frac{1}{\sqrt{N}}$, i.e.

$$\forall n \in [0, N-1] \quad \mathbf{x}(n) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \delta_m e^{2\pi imn/N}$$

We define δ_m to be the m-th component of the **discrete Fourier transform** of $\mathbf{x}$, and the operation that reconstructs $\mathbf{x}(n)$ is called **inverse discrete Fourier transform**. In other words, the DFT of a vector computes the projection of said vector into the space $\mathbb{C}^N$ through the orthonormal basis

$$\left\{ \tfrac{1}{\sqrt{N}} e^{2\pi imn/N} \right\}_{m=0}^{N-1}$$

Indeed, we can compute the DFT by simply applying the matrix that performs the **change of basis**, but we will see the details of this idea in the next section.

4.2 The Quantum Fourier Transform

Now that we have a general understanding of what the DFT is, let's see how the matrix operation is actually defined.

Definition 4.1: n-th root of unity

Given a commutative ring R, and a natural number $n \in N$, an element $\omega \in R$ is called **n-th root of unity** if $\omega^n = 1$.

For instance, $-i$ is a 4-th root of unity, in fact $(-i)^4 = 1$. We observe that, since multiplication in the complex plane is a rotation, the $n-$-th roots of 1 evenly divide the unit circle as shown below:

TODO drawing

Now, among all possible n-th roots of unity we are going to provide a more specific definition. Let the **order of an n-th root of unity** ω be the smallest power d such that $\omega^d = 1$. For instance, $(-1)^4 = 1$ indeed -1 is a 4-th root of unity, however its order is 2 since $(-1)^2 = 1$ and $2 < 4$.

Definition 4.2: Principal n-th root of unity

Given an n-th root of unity ω, we say that ω is **principal** if and only if $\omega \neq 1$ and the order of ω is n.

In other words, the principal n-th root of unity is the first root (after 1) that we encounter on the unit circle. Indeed, thanks to Euler's formula we usually define the principal n-th principal root of unity as

$$\omega := e^{2\pi i/n}$$

since $\frac{2\pi}{n}$ is the n-th slice of the unit circle. Furthermore, the second condition of the definition is usually not provided in terms of order of the root, and it's written as follows

$$\forall p \in [n-1] \quad \sum_{j=0}^{n-1} \omega^{jp} = 0$$

Aside from the geometric interpretation of this sum, this is a finite complex geometric series with common ratio ω^p, which implies that

- if $\omega^p = 1$, then every term is $(\omega^p)^j = 1^j = 1$ for any j, meaning that the sum is equal to
$$\sum_{j=0}^{n-1} \omega^{jp} = \sum_{j=0}^{n-1} 1 = n$$
- if $\omega^p \neq 1$, then we know that
$$\sum_{j=0}^{n-1} \omega^{jp} = \frac{1 - (\omega^p)^n}{1 - \omega^p}$$
and this ratio is equal to 0 if and only if
$$1 - (\omega^p)^n = 0 \iff \omega^{pn} = 1$$

Therefore, we have that this sum is equal to 0 if and only if $\omega^p \neq 1$ and $\omega^{pn} = 1$. However, we observe that if $\omega^p = 1$, since $p \in [n-1]$ this would imply that the order of ω is less than n, meaning that ω was not a principal root.

Now that we presented roots of unity we can finally present how the DFT matrix is usually defined.

Definition 4.3: Discrete Fourier Transform

Given a commutative ring R of dimension N, the **Discrete Fourier Transform (DFT)** matrix in R is defined as follows:

$$\forall j, k \in [0, N-1] \quad \mathrm{DFT}_{jk} = \omega^{-jk}$$

where ω is the principal N-th root of unity.

Therefore, in general the DFT matrix looks like this:

$$\mathrm{DFT} := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(N-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \cdots & \omega^{-(N-1)^2} \end{pmatrix}$$

Moreover, we usually define the DFT matrix by setting $\omega = e^{\frac{2\pi i}{n}}$, thus getting

$$\forall j, k \in [0, N-1] \quad \mathrm{DFT}_{jk} = e^{-2\pi i jk/N}$$

Indeed, having presented the intuition behing the DFT beforehand, it's now fairly obvious the reason why the DFT matrix looks like this: its just the matrix that performs the change of basis, and since the basis of the DFT space can be defined in terms of n-th roots of unity, this is the matrix we get.

Furthermore, not surprisingly the DFT matrix is invertible, in fact we have that

$$\forall j, k \in [0, N-1] \quad \mathrm{IDFT}_{jk} = \mathrm{DFT}^{-1}_{jk} = \frac{1}{N}\omega^{jk} = \frac{1}{N}e^{+2\pi i jk/N}$$

and this matrix takes the name of **Inverse Discrete Fourier Transform (IDFT)**.

Then, can't we just use the DFT matrix in quantum computations whenever we need it and be done? The problem is that the DFT matrix is clearly not unitary: if we look closely, we see that the columns of the DFT matrix are only orthogonal and not orthonormal. Thus, to solve this problem, we need to normalize the column vectors.

Definition 4.4: Quantum Fourier Transform

Given a Hilbert space $\mathcal{H}$ of size N, the **Quantum Fourier Transform (QFT)** matrix in $\mathcal{H}$ is defined as follows:

$$\forall j, k \in [0, N-1] \quad \mathrm{QFT}_{jk} = \frac{1}{\sqrt{N}}\omega^{jk}$$

where ω is the principal N-th root of unity.

We observe that the QFT matrix has positive exponents, which is just a convention employed in quantum computing. Therefore, we get the following matrix:

$$\mathrm{QFT} := \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)^2} \end{pmatrix}$$

In particular, for any $|x\rangle$ basis state, the quantum Fourier transform can also be expressed as follows:

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{+2\pi i x k/N}\,|k\rangle$$

which directly implies that we can rewrite the QFT matrix as follows

$$\mathrm{QFT} = \sum_{j=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N}\,|k\rangle \right) \langle j|$$

thanks to Proposition 2.10.

Proposition 4.1

The QFT operator is a unitary, and in particular

$$\mathrm{QFT}^\dagger = \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{-2\pi i j k/N}\,\langle k| \right)$$

Proof. We will leave the proof of correctness of $\mathrm{QFT}^\dagger$ as an exercise, and we will focus on proving that the QFT operator is indeed unitary. To prove it, is satisfies to show that

$$\mathrm{QFT}^\dagger\ \mathrm{QFT} = \mathrm{QFT}\ \mathrm{QFT}^\dagger = I$$

We are going to prove the leftmost product first.

$$\begin{aligned}
\mathrm{QFT}^\dagger\ \mathrm{QFT} &= \sum_{j=0}^{N-1} |j\rangle \left(\frac{1}{\sqrt{N}} \sum_{r=0}^{N-1} e^{-2\pi i j r/N} \langle r|\right) \sum_{k=0}^{N-1} \left(\frac{1}{\sqrt{N}} \sum_{s=0}^{N-1} e^{2\pi i s k/N} |s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{-2\pi i j r/N} \langle r|\right) \left(\sum_{s=0}^{N-1} e^{2\pi i s k/N} |s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r,s=0}^{N-1} e^{2\pi i(-jr+ks)/N} \langle r|s\rangle\right) \langle k| \\
&= \frac{1}{N} \sum_{j,k=0}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N}\right) \langle k| && (\langle r|s\rangle = \delta_{rs}) \\
&= \frac{1}{N} \sum_{\substack{j,k=0: \\ j=k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} 1\right) \langle k| + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N}\right) \langle k| \\
&= \frac{1}{N} \sum_{j=0}^{N-1} |j\rangle \langle j| + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N}\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} e^{2\pi i r(k-j)/N}\right) \langle k| && (\text{resolution of } I) \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} \left(e^{2\pi i(k-j)/N}\right)^r\right) \langle k| && (\text{resolution of } I)
\end{aligned}$$

Now, we observe that the inner sum is a geometric series of ratio $e^{2\pi i(k-j)/N}$, and we can use its known result to simplify the calculations. However, we observe that this is true only if the ratio of the series is not equal to 1, but due to the way we split the sums we already took care of all the possible terms that could be equal to 1 so we do not need to make additional assumptions. Thus, we have that

$$\begin{aligned}
\mathrm{QFT}^\dagger\ \mathrm{QFT} &= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\sum_{r=0}^{N-1} \left(e^{2\pi i(k-j)/N}\right)^r\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\frac{1 - \left(e^{2\pi i(k-j)/N}\right)^N}{1 - e^{2\pi i(k-j)/N}}\right) \langle k| \\
&= I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j\neq k}}^{N-1} |j\rangle \left(\frac{1 - e^{2\pi i(k-j)}}{1 - e^{2\pi i(k-j)/N}}\right) \langle k|
\end{aligned}$$

Lastly, since $k - j$ is an integer for any $j, k \in [0, N-1]$ — and in particular whenever $j \neq k$ — it holds that

$$e^{2\pi i(k-j)} = \cos(2\pi i(k-j)) + i \sin(2\pi i(k-j)) = 1$$

Therefore, the numerator is always equal to $1 - 1 = 0$, so the sum vanishes. However, we need to check that the denominator is well defined too: we observe that

$$e^{2\pi i(k-j)/N} = \cos(2\pi i(k-j)) + i\sin(2\pi i(k-j))$$

and this is never equal to 1 because $(k-j)/N$ is always not an integer since $j \neq k$ due to how we split the sums. Finally, we can conclude that

$$\mathrm{QFT}^\dagger \ \mathrm{QFT} = I + \frac{1}{N} \sum_{\substack{j,k=0: \\ j \neq k}}^{N-1} |j\rangle \left(\frac{1-1}{1 - e^{2\pi i(k-j)/N}} \right) \langle k| = I$$

TODO

prove the second product

□

4.2.1 The quantum circuit

The previous section showed how the QFT matrix operator is defined, why its unitary and its similarities with the DFT. It's finally time to address the most important question: how do we turn the QFT matrix into a quantum circuit? Assume that $N = 2^n$, and consider the definition we provided

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{+2\pi i x k/N} \,|k\rangle$$

The problem arises because we are using fractional exponents, however $x \in \mathbb{B}^n$. To solve this issue, we need to take a closer look at the exponent. Since $x \in \mathbb{B}^n$, there exist some bits $x_1, \ldots, x_n \in \mathbb{B}$ such that x can be written as

$$x = x_1 \cdot 2^{n-1} + \ldots + x_n \cdot 2^0$$

Since $k \in [0, N-1]$, we can also define such $k_1, \ldots, k_n \in \mathbb{B}$, and we can see what happens when we evaluate xk/N:

$$\begin{aligned} kx/N &= \frac{(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0) \cdot (x_1 2^{n-1} + \ldots + x_n 2^0)}{2^n} \\ &= \left(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0\right) \cdot \left(x_1 \cdot \frac{2^{n-1}}{2^n} + \ldots + x_n \cdot \frac{2^0}{2^n} \right) \\ &= \left(k_1 \cdot 2^{n-1} + \ldots + k_n \cdot 2^0\right) \cdot 0.x_1 \ldots x_n \\ &= k_1 \cdot 2^{n-1} \cdot 0.x_1 \ldots x_n + \ldots + k_n \cdot 2^0 \cdot 0.x_1 \ldots x_n \end{aligned}$$

This is true because we recall that

$$0.x_1 \ldots x_n = x_1 \cdot 2^{-1} + \ldots + x_n \cdot 2^{-n}$$

and since we are computing x/N and $N = 2^n$ what we get is just a "displacement of the decimal point", but in binary. Now, we observe that

$$\forall \ell \in [n] \quad 2^{n-\ell} \cdot 0.x_1 \ldots x_n = x_1 \ldots x_{n-\ell}.x_{n-\ell+1} \ldots x_n$$

since again, we are just moving the decimal point $n - \ell$ times to the right. Therefore, we get that

$$\begin{aligned} kx/N &= k_1 \cdot x_1 \dots x_{n-1}.x_n + \dots + k_n 0.x_1 \dots x_n \\ &= k_n \cdot 0.x_1 \dots x_n + \sum_{h=1}^{n-1} k_h \cdot x_1 \dots x_{n-h}.x_{n-h+1} \dots x_n \end{aligned}$$

we observe that we need to put $0.x_1 \dots x_n$ just because we cannot include it in the same sum. Now, let's see what happens when we put this term at the exponent:

$$\begin{aligned} & e^{2\pi i k x/N} \\ &= \exp(2\pi i k x/N) \\ &= \exp\left(2\pi i \left(k_n 0.x_1 \dots x_n + \sum_{h=1}^{n-1} k_h \cdot x_1 \dots x_{n-h}.x_{n-h+1} \dots x_n\right)\right) \\ &= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot x_1 \dots x_{n-h}.x_{n-h+1} \dots x_n\right) \\ &= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot (x_1 \dots x_{n-h}) + 2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right) \\ &= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot (x_1 \dots x_{n-h})\right) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right) \\ &= \exp(2\pi i k_n 0.x_1 \dots x_n) \cdot 1 \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right) \end{aligned}$$

The last semplitifcation derives immediately from the fact that $e^{2\pi i\theta} = 1$ when θ is an integer, as we mentioned in the previous section, so that is what we are left with — since $x_1 \dots x_{n-h}$ is definitely an integer. We can proceed as follows:

$$\begin{aligned} & \mathrm{QFT}\,|x\rangle \\ =& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{h=1}^{n-1} k_h \cdot 0.x_{n-h+1} \dots x_n\right)\right) |k\rangle \\ =& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(\exp(2\pi i k_n 0.x_1 \dots x_n) \cdot \exp\left(2\pi i \sum_{m=2}^{n} k_{n-m+1} \cdot 0.x_m \dots x_n\right)\right) |k\rangle \\ =& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(2\pi i \sum_{m=1}^{n} k_{n-m+1} \cdot 0.x_m \dots x_n\right) |k\rangle \\ =& \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \prod_{m=1}^{n} \exp\left(2\pi i k_{n-m+1} \cdot 0.x_m \dots x_n\right) |k\rangle \end{aligned}$$

If we look closely, this sum of products as an interesting property:

- $m = 1 \implies k_{n-m+1} = k_{n-1+1} = k_n$

- $m = n \implies k_{n-m+1} = k_{n-n+1} = k_1$

implying that it can be rewritten as follows

$$\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \left(e^{2\pi i k_n \cdot 0.x_1 \ldots x_n} \cdot \ldots \cdot e^{2\pi i k_1 \cdot 0.x_n} \right)$$

In other words the bits of k are *indexing* which exponentials to use in the k-th coefficient of the sum. This means that we can rewrite the last step as follows:

$$\mathrm{QFT} \left| x \right\rangle = \frac{1}{\sqrt{N}} \bigotimes_{m=1}^{n} \left(\left| 0 \right\rangle + e^{2\pi i 0.x_m \ldots x_n} \left| 1 \right\rangle \right)$$

Finally, the QFT written in this form can be actually implemented in a quantum circuit.

da finire sonostufo

4.3 Quantum Phase Estimation

Let's see a very useful application of the QFT introduced by Kitaev [Kit95] in 1995. By Theorem 2.2 we know that the eigenvalues of a unitary operator are complex values of modulus 1. This means that any eigenvalue of a unitary operator can be ewritten as $e^{2\pi i \varphi}$ for some real value $\varphi \in [0, 1]$ — φ being the *phase* of the complex value, i.e. the angle w.r.t. the unitary circumference.

Given a unitary operator U, and an eigenvalue $\lambda = e^{2\pi i \varphi}$, can we determine φ? Let u be an eigenvector associated to the unknown eigenvalue λ, i.e.

$$U \left| u \right\rangle = \lambda \left| u \right\rangle = e^{2\pi i \varphi} \left| u \right\rangle$$

Now, let U^{2^k} be the quantum gate that applies the U operator 2^k times repeatedly, and define the following operator

$$\text{C-}U^{2^k} = \left| 0 \right\rangle \left\langle 0 \right| \otimes I + \left| 1 \right\rangle \left\langle 1 \right| \otimes U^{2^k}$$

This operator is the **controlled** version of U^{2^k}, and it can easily be showed by plugging $\left| 0 \right\rangle$ and $\left| 1 \right\rangle$ as first argument — alongside with some other quantum state $\left| \psi \right\rangle$

$$\begin{aligned} \text{C-}U^{2^k} (\left| 0 \right\rangle \otimes \left| \psi \right\rangle) &= (\left| 0 \right\rangle \left\langle 0 \right| \otimes I + \left| 1 \right\rangle \left\langle 1 \right| \otimes U^{2^k})(\left| 0 \right\rangle \otimes \left| \psi \right\rangle) \\ &= (\left| 0 \right\rangle \left\langle 0 \right| \otimes I) \left| 0 \right\rangle \otimes \left| \psi \right\rangle + (\left| 1 \right\rangle \left\langle 1 \right| \otimes U^{2^k}) \left| 0 \right\rangle \otimes \left| \psi \right\rangle \\ &= \left| 0 \right\rangle \otimes \left| \psi \right\rangle \end{aligned}$$

Hence, if the first input is $\left| 0 \right\rangle$ the state $\left| \psi \right\rangle$ is unchanged, otherwise if the former is $\left| 1 \right\rangle$ we get that we actually apply U^{2^k} to $\left| \psi \right\rangle$:

$$\begin{aligned} \text{C-}U^{2^k} (\left| 1 \right\rangle \otimes \left| \psi \right\rangle) &= (\left| 0 \right\rangle \left\langle 0 \right| \otimes I + \left| 1 \right\rangle \left\langle 1 \right| \otimes U^{2^k})(\left| 1 \right\rangle \otimes \left| \psi \right\rangle) \\ &= (\left| 0 \right\rangle \left\langle 0 \right| \otimes I) \left| 1 \right\rangle \otimes \left| \psi \right\rangle + (\left| 1 \right\rangle \left\langle 1 \right| \otimes U^{2^k}) \left| 0 \right\rangle \otimes \left| \psi \right\rangle \\ &= \left| 1 \right\rangle \otimes U^{2^k} \left| \psi \right\rangle \end{aligned}$$

Indeed, in general if we have an operator V, its controlled version can be easily built in this manner.

However, this controlled gate can also be "abused": what happens if the control bit is a superposition of states? In particular, what happens when the control is set to

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

meaning that $|0\rangle$ and $|1\rangle$ are equally probable?

$$\begin{aligned}
\text{C-}U^{2^k}(|+\rangle \otimes |\psi\rangle) &= \frac{1}{\sqrt{2}}\text{C-}U^{2^k}(|0\rangle \otimes |\psi\rangle + |1\rangle \otimes |\psi\rangle) \\
&= \frac{1}{\sqrt{2}}(\text{C-}U^{2^k}(|0\rangle \otimes |\psi\rangle) + \text{C-}U^{2^k}(|1\rangle \otimes |\psi\rangle)) \\
&= \frac{1}{\sqrt{2}}((|0\rangle \otimes |\psi\rangle) + (|1\rangle \otimes U^{2^k}|\psi\rangle))
\end{aligned}$$

As we would expect, what happens is that the two outcomes we previously described are now equally likely, but this gets interesting if we carefully choose the target qubit. Instead of any possible $|\psi\rangle$, let's pick $|u\rangle$, the eigenvector presented at the beginning of the discussion. First, we observe that $U|u\rangle = e^{2\pi i\varphi}|u\rangle \implies U^{2^k}|u\rangle = (e^{2\pi i\varphi})^{2^k}|u\rangle = e^{2\pi i 2^k\varphi}|u\rangle$ Indeed, intuitively we are just performing the rotation that U performs on $|u\rangle$ exactly 2^k times. This means that

$$\begin{aligned}
\text{C-}U^{2^k}(|+\rangle \otimes |u\rangle) &= \frac{1}{\sqrt{2}}((|0\rangle \otimes |u\rangle) + (|1\rangle \otimes U^{2^k}|u\rangle)) \\
&= \frac{1}{\sqrt{2}}((|0\rangle \otimes |u\rangle) + (|1\rangle \otimes e^{2\pi i 2^k\varphi}|u\rangle)) \\
&= \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^k\varphi}|1\rangle) \otimes |u\rangle
\end{aligned}$$

Notice what happened here: $|0\rangle$ and $|1\rangle$ are parts of the first input, and $|u\rangle$ is the second input. In other words, by putting $|u\rangle$ as target input of the controlled gate, the target is **unchanged**, and the effect is seen on the control. We observe that this could not have been done with any other state $|\psi\rangle$ because the trick works precisely because U becomes a scalar $e^{2\pi i\varphi}$ when applied to $|u\rangle$, so it can be grouped as shown. The following quantum circuit computes the **Quantum Phase Estimation (QPE)** algorithm, and it uses precisely this trick.

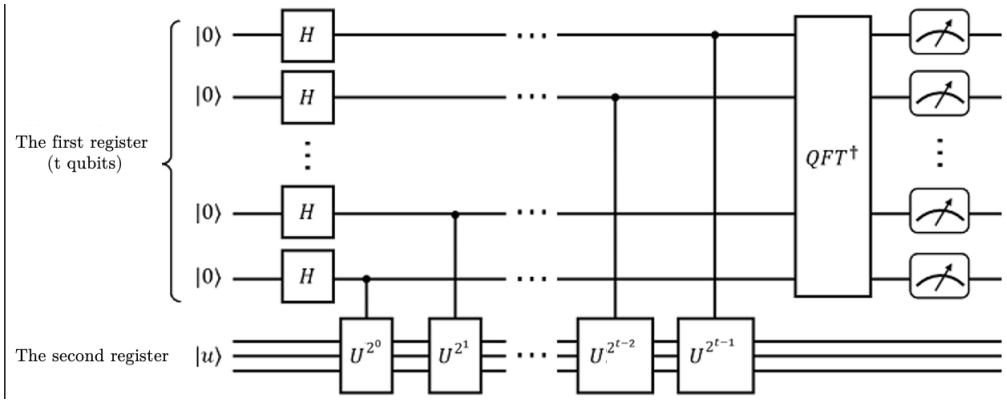


Figure 4.1: The Quantum Phase Estimation circuit.

The circuit is composed of the following elements:

fixa l'immagine che la misurazione non va bene scritta cosi

- the first are $t > 0$ qubits $|0^t\rangle$ to which we apply $H^{\otimes t}$ at the beginning of the algorithm, which means now each state is set to $|+\rangle$
- the $(t+1)$-th register is a qubit set to $|u\rangle$ — we are not going to cover the practical details about how to set a qubit to this precise quantum state
- these superpositions are used as control to the lower part of the circuit, which conditionally computes U^{2^k} for each $k \in [0, t-1]$
- at the end we compute $\text{QFT}^\dagger$ to the first t registers

The interesting part of this circuit is that the last register remains unchanged throughout the whole computation, set precisely to $|u\rangle$, for the reason we described earlier. In fact, what happens is that what we are actually changing are the first t qubits which will be chaned according to the k-th controlled gate we are using. Therefore, if we compute what happens through the QPE circuit we get the following — we will refer to the first t

registers as $q_0, \ldots, q_{t-1}$:

$$
\begin{aligned}
&|q_0 \cdots q_{t-1}\rangle \otimes |u\rangle \\
=&|0\rangle^{\otimes t} \otimes |u\rangle \\
\xrightarrow{H(q_0)}&H(|0\rangle_{q_0}) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&|+\rangle_{q_0} \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&|+\rangle_{q_0} \otimes |u\rangle \otimes |0\rangle^{\otimes t-1} \\
\xrightarrow{\text{C-}U^{2^0}(q_0,|u\rangle)}&\text{C-}U^{2^0}(|+\rangle \otimes |u\rangle) \otimes |0\rangle^{\otimes t-1} \\
&\frac{1}{\sqrt{2}}(|0\rangle_{q_0} + e^{2\pi i 2^0 \varphi}|1\rangle_{q_0}) \otimes |u\rangle \otimes |0\rangle^{\otimes t-1} \\
=&\frac{1}{\sqrt{2}}(|0\rangle_{q_0} + e^{2\pi i 2^0 \varphi}|1\rangle_{q_0}) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
=&\frac{1}{\sqrt{2}}(|0\rangle_{q_0} \otimes |0\rangle^{\otimes t-1} + e^{2\pi i 2^0 \varphi}|1\rangle_{q_0} \otimes |0\rangle^{\otimes t-1}) \otimes |u\rangle \\
\xrightarrow{H(q_1)}&\frac{1}{\sqrt{2}}(|0\rangle_{q_0} \otimes H(|0\rangle_{q_1}) \otimes |0\rangle^{\otimes t-2} + e^{2\pi i 2^0 \varphi}|1\rangle_{q_0} \otimes H(q_1) \otimes |0\rangle^{\otimes t-2}) \otimes |u\rangle \\
=&\frac{1}{2}\left(|00\rangle_{q_0q_1} + |01\rangle_{q_0q_1} + e^{2\pi i 2^0 \varphi}|10\rangle_{q_0q_1} + e^{2\pi i 2^0 \varphi}|11\rangle_{q_0q_1}\right) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
\xrightarrow{\text{C-}U^{2^1}(q_1,|u\rangle)}&\frac{1}{2}\left(|00\rangle_{q_0q_1} + e^{2\pi i 2^1 \varphi}|01\rangle_{q_0q_1} + e^{2\pi i 2^0 \varphi}|10\rangle_{q_0q_1} + e^{2\pi i (2^0+2^1) \varphi}|11\rangle_{q_0q_1}\right) \otimes |0\rangle^{\otimes t-1} \otimes |u\rangle \\
\xrightarrow{\cdots}&\cdots \\
=&\frac{1}{\sqrt{2^t}}\sum_{k=0}^{2^t-1}\prod_{m=1}^{t} e^{2\pi i k_{t-m+1} \cdot 2^{m-1} \varphi}|k\rangle \otimes |u\rangle \\
=&\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 2^{m-1} \varphi}|1\rangle\right) \otimes |u\rangle
\end{aligned}
$$

Furthermore, we obseve that $\varphi \in [0, 1)$, because as we already metioned there is no need to consider the integral part of the phases. Thus, suppose that φ can be written through t bits $\varphi_1, \ldots, \varphi_t \in \mathbb{B}$ such that

$$\varphi = 0.\varphi_1 \ldots \varphi_t$$

This implies that

$$
\begin{aligned}
2^{m-1} \cdot \varphi &= 2^{m-1} \cdot 0.\varphi_1 \ldots \varphi_t \\
&= \varphi_1 \ldots \varphi_{m-1}.\varphi_m \ldots \varphi_t
\end{aligned}
$$

Thus, since this is the exponent of $e^{2\pi i}$, we already know that

$$\exp(2\pi i \varphi_1 \ldots \varphi_{m-1}.\varphi_m \ldots \varphi_t) = \exp(2\pi i 0.\varphi_m \ldots \varphi_t)$$

therefore the ternsor product can be rewritten as

$$\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}(|0\rangle + e^{2\pi i 0.\varphi_m \ldots \varphi_t}|1\rangle)$$

Very elegantly, this is exactly $\mathrm{QFT}_t |\varphi\rangle$ we computed in the previous section (only applied in a t-dimensional space). Finally, if we place the inverse QFT — namely $\mathrm{QFT}_t^\dagger$ — at the end of the circuit we retrieve $|x\rangle$ itself, which is also the entire state of the system at this point (without considering $|u\rangle$). This means that if we measure each register at the end of the circuit we retrieve the bits of $\varphi_i \in \mathbb{B}^t$ — i.e. the bits that describe φ — with probability 1. In the end, we recovered the phase φ of the eigenvalue associated to $|u\rangle$ with probability 1.

scrivere QPE come alg

As a final note, what if φ cannot be expressed in exactly t bits? It can be shown that φ can be estimated with n bits of precision through a QPE circuit composed of $n + \left\lceil \log(2 + \frac{1}{2\varepsilon}) \right\rceil$ qubits with a success probability at least $1 - \varepsilon$, for some $\varepsilon > 0$.

4.4 Order-finding problem

The last ingredient that we need for Shor's algorithm is the **order-finding** algorithm. We recall the following property of number theory.

Proposition 4.2: Euclidean division

Given some $p \in \mathbb{N}_{>0}$, for any $n \in \mathbb{N}$ there exists unique integers $q, r \in \mathbb{Z}$ such that $0 \leq r < p$ and

$$n = p \cdot q + r$$

This is the usual division with remainder, which defines modular arithmetic, for instance

$$31 = 4 \cdot 7 + 3$$

Indeed, with equivalence classes we would write that

$$31 \equiv 3 \bmod 7$$

Definition 4.5: Order of a number

Given a number $N \in \mathbb{N}$, for any $x \in \mathbb{Z}$ the **order** of x modulo N is the least integer r such that

$$x^r \equiv 1 \bmod N$$

For instance, the order of 4 modulo 7 is 3, because

$$4^3 = 64 = 9 \cdot 7 + 1 \implies 4^3 \equiv 1 \bmod 7$$

We observe that throughout our discussion N is *not necessarily* a power of 2 as we used to assume in previous sections, however the symbol N is conventionally used in this context.

Given a number N, and a number $x < N$ that is coprime with N, can we compute its order modulo N? This is the so called **order-finding problem**, and currently there

is no classical algorithm able to solve it in polynomial time. Let's see what quantum computation can achieve.

Consider the following unitary operator U_x that computes as follows:

$$\forall y \in \{0,1\}^k \quad U_x |y\rangle = \begin{cases} |xy \bmod N\rangle & y < N \\ |y\rangle & y \geq N \end{cases}$$

To derive the complete operator U_x we just follow Proposition 2.10:

$$\begin{aligned} U_x &= \sum_{y \in \{0,1\}^k} U_x |y\rangle \langle y| \\ &= \sum_{y<N} U_x |y\rangle \langle y| + \sum_{y \geq N} U_x |y\rangle \langle y| \\ &= \sum_{y<N} U_x |y\rangle \langle y| + \sum_{y \geq N} |y\rangle \langle y| \end{aligned}$$

We will not replace U_x inside the first sum for now in order to prove that U_x is unitary. But first, we need to present a result in number theory.

Theorem 4.1

If x and N are coprime, it holds that

$$\forall y, z \in \mathbb{Z} \quad xy \equiv xz \bmod N \iff y \equiv z \bmod N$$

Proposition 4.3

The operator U_x is unitary, and in particular

$$U_x^\dagger = \sum_{y<N} |y\rangle (U_x |y\rangle)^\dagger + \sum_{y \geq N} |y\rangle \langle y|$$

Proof. It is easy to prove the correctness of $U_x^\dagger$, so we are going to prove that U_x is unitary directly.

$$\begin{aligned} U_x^\dagger U_x &= \left(\sum_{y<N} |y\rangle (U_x |y\rangle)^\dagger + \sum_{y \geq N} |y\rangle \langle y| \right) \left(\sum_{z<N} U_x |z\rangle \langle z| + \sum_{z \geq N} |z\rangle \langle z| \right) \\ &= \left(\sum_{y<N} |y\rangle \langle U_x y| + \sum_{y \geq N} |y\rangle \langle y| \right) \left(\sum_{z<N} |U_x z\rangle \langle z| + \sum_{z \geq N} |z\rangle \langle z| \right) \\ &= \sum_{y,z<N} |y\rangle \langle U_x y | U_x z\rangle \langle z| + \sum_{\substack{y<N \\ z \geq N}} |y\rangle \langle U_x y | z\rangle \langle z| + \sum_{\substack{y \geq N \\ z<N}} |y\rangle \langle y | U_x z\rangle \langle z| + \sum_{y,z \geq N} |y\rangle \langle y | z\rangle \langle z| \end{aligned}$$

Now we observe that if $z > N$ and $y \leq N$, by definition of U_x we have that $U_x |y\rangle = |xy \bmod N\rangle$, hence it will be a basis state among $|0\rangle, \ldots, |N-1\rangle$. Therefore, if $z > N$

we are guaranteed that $|z\rangle$ and $|U_x y\rangle$ are orthogonal, i.e. $\langle U_x y|z\rangle = 0$ — we recall that N is *not* the size of the space in this context, is just a composite number, indeed the size of the space we are considering is 2^k. Therefore, we have that

$$\sum_{\substack{y<N\\z\geq N}} |y\rangle \langle U_x y|z\rangle \langle z| = \sum_{\substack{y\geq N\\z<N}} |y\rangle \langle y|U_x z\rangle \langle z| = 0$$

so we conclude that

$$\begin{aligned}
U_x^\dagger U_x &= \sum_{y,z<N} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{\substack{y<N\\z\geq N}} |y\rangle \langle U_x y|z\rangle \langle z| + \sum_{\substack{y\geq N\\z<N}} |y\rangle \langle y|U_x z\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{y,z<N} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{\substack{y,z<N:\\ y\not\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z|
\end{aligned}$$

To progress, we observe that when $y, z \leq N$ we get that

$$\langle U_x y|U_x z\rangle = \langle xy \bmod N|xz \bmod N\rangle = \begin{cases} 1 & xy \equiv xz \bmod N \\ 0 & \text{otherwise} \end{cases}$$

and by Theorem 4.1 we know that

$$xy \equiv xz \bmod N \iff y \equiv z \bmod N$$

thus getting

$$\begin{aligned}
U_x^\dagger U_x &= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{\substack{y,z<N:\\ y\not\equiv z \bmod N}} |y\rangle \langle U_x y|U_x z\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{\substack{y,z<N:\\ y\equiv z \bmod N}} |y\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \langle y|z\rangle \langle z| \\
&= \sum_{\substack{y,z<N:\\ y=z}} |y\rangle \langle z| + \sum_{y,z\geq N} |y\rangle \delta_{yz} \langle z| \\
&= \sum_{y<N} |y\rangle \langle y| + \sum_{y\geq N} |y\rangle \langle y| \\
&= \sum_{y} |y\rangle \langle y| \\
&= I
\end{aligned}$$

Now, we need to prove the opposite product □ da fare

We did not provide any reason to why we defined the operator U_x as such, but before giving a geometrical intuition consider the following proposition.

Theorem 4.2

Given $N \in \mathbb{N}$, and $x \in [0, N-1]$, if r is the order of x modulo N, it holds that

$$\forall s \in [0, r-1] \quad |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x^k \bmod N\rangle$$

is an eigenvector of U_x.

Proof. Fix $s \in [0, r-1]$; to prove that $|u_s\rangle$ is an eigenvector of U_x we need to show that there exists some phase φ_s such that

$$U_x \, |u_s\rangle = e^{2\pi i \varphi_s} \, |u_s\rangle$$

because of Theorem 2.2. Hence, we get that

$$\begin{aligned}
U_x \, |u_s\rangle &= U_x \left(\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x^k \bmod N\rangle \right) \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} U_x \, |x^k \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x \left(x^k \bmod N\right) \bmod N\rangle && \left(x^k \bmod N < N\right) \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x^{k+1} \bmod N \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \cdot \frac{e^{2\pi i s/r}}{e^{2\pi i s/r}} \cdot \sum_{k=0}^{r-1} e^{-2\pi i s k/r} \, |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \cdot \sum_{k=0}^{r-1} e^{-2\pi i s (k+1)/r} \, |x^{k+1} \bmod N\rangle \\
&= \frac{1}{\sqrt{r}} \cdot e^{2\pi i s/r} \left(\sum_{k=0}^{r-2} e^{-2\pi i s (k+1)/r} \, |x^{k+1} \bmod N\rangle + e^{-2\pi i s r/r} \, |x^r \bmod N\rangle \right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{k=0}^{r-2} e^{-2\pi i s (k+1)/r} \, |x^{k+1} \bmod N\rangle + e^{-2\pi i s} \, |1\rangle \right) && \left(x^r \equiv 1 \bmod N\right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{m=1}^{r-1} e^{-2\pi i s m/r} \, |x^m \bmod N\rangle + |1\rangle \right) \\
&= \frac{1}{\sqrt{r}} e^{2\pi i s/r} \left(\sum_{m=0}^{r-1} e^{-2\pi i s m/r} \, |x^m \bmod N\rangle \right) && \left(|1\rangle = |x^0 \bmod N\rangle\right) \\
&= e^{2\pi i s/r} \, |u_s\rangle
\end{aligned}$$

Therefore, we conclude that $\varphi_s = s/r$. □

TODO

intuizione geometrica

What happenns if the input of U_x is $|x^0\rangle$? By definition of our problem $x < N$, therefore

$$U_x |x^0\rangle = |x \cdot x^0 \bmod N\rangle = |x^1 \bmod N\rangle$$

Indeed in general it's easy to see that

$$U_x |x^k\rangle = |x^{k+1} \bmod N\rangle$$

In other words U_x is cycling through x's powers, which also implies that when $k = r - 1$ we get that

$$U_x |x^{r-1}\rangle = |x^r \bmod N\rangle = |1 \bmod N\rangle$$

since r is the order of x modulo N. Moreover, as we already know these are basis states so they are both normalized and orthogonal to each other, thus the powers x form an orthonormal base of the following space

$$\mathcal{H}_r = \text{span}\left(|x^k \bmod N\rangle \mid k \in [0, r-1]\right)$$

which is a restriction of the whole Hilbert space that has exactly r dimensions. Now look again at how the eigenvectors of U_x are defined:

$$\forall s \in [0, r-1] \quad |u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i s k/r} |x^k \bmod N\rangle$$

This looks very similar to how we defined $\text{QFT}\,|x\rangle$:

$$\text{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/N} |k\rangle$$

Indeed, it holds that $|u_s\rangle$ is basically $\text{QFT}\,|s\rangle$ but taken inside $\mathcal{H}_r$ — the only difference being the sign of the exponent, ndeed it technically holds that

$$|u_s\rangle = \text{QFT}_r\,|-s \bmod r\rangle$$

but as we said the sign of the exponent is just a convention either sign is found in literature, we only need to be consistent with the calculations.

non ho capito che c'entra però

The most interesting part is that, as proved in the last theorem, for any fixed $s \in [0, N-1]$ its associated phase φ_s is exactly s/r, thus if we knew how to prepare the last register of the QPE as $|u_s\rangle$ we could recover its phase, and maybe get closer to know r itself. However, we have two probems with this idea:

- there is really no easy or practical way to prepare the second register to some arbitrary state
- $|u_s\rangle$ actually depends on r itself, which actually is a very big problem on its own

Thankfully, the following property will solve both of these issues at the same time.

Proposition 4.4

Given $N \in \mathbb{N}$, and $x \in [0, N-1]$, if r is the order of x modulo N it holds that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle$$

Proof. Through some algebraic manipulation we see that

$$\begin{aligned}
\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle &= \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} \left(\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi isk/r} |x^k \bmod N\rangle \right) \\
&= \frac{1}{r} \sum_{s,k=0}^{r-1} e^{-2\pi isk/r} |x^k \bmod N\rangle \\
&= \frac{1}{r} \sum_{s=0}^{r-1} e^{-2\pi is\cdot 0/r} |x^0 \bmod N\rangle + \frac{1}{r} \sum_{s=0,k=1}^{r-1} e^{-2\pi isk/r} |x^k \bmod N\rangle \\
&= \frac{1}{r} \sum_{s=0}^{r-1} |1\rangle + \frac{1}{r} \sum_{s=0,k=1}^{r-1} e^{-2\pi isk/r} |x^k \bmod N\rangle \\
&= \frac{1}{r} r |1\rangle + \frac{1}{r} \sum_{s=0,k=1}^{r-1} e^{-2\pi isk/r} |x^k \bmod N\rangle \\
&= |1\rangle + \frac{1}{r} \sum_{k=1}^{r-1} |x^k \bmod N\rangle \sum_{s=0}^{r-1} e^{-2\pi isk/r} \\
&= |1\rangle + \frac{1}{r} \sum_{k=1}^{r-1} |x^k \bmod N\rangle \sum_{s=0}^{r-1} \left(e^{-2\pi ik/r}\right)^s
\end{aligned}$$

As we did for the proof of Proposition 4.1, because of how we split the sums we know that $e^{-2\pi ik/r}$ cannot be 0, but it cannot be 1 either because k/r is not an integer, therefore we get that

$$\begin{aligned}
\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle &= |1\rangle + \frac{1}{r} \sum_{k=1}^{r-1} |x^k \bmod N\rangle \frac{1 - (e^{-2\pi ik/r})^r}{1 - e^{-2\pi ik/r}} \\
&= |1\rangle + \frac{1}{r} \sum_{k=1}^{r-1} |x^k \bmod N\rangle \frac{1 - e^{-2\pi ik}}{1 - e^{-2\pi ik/r}} \\
&= |1\rangle + \frac{1}{r} \sum_{k=1}^{r-1} |x^k \bmod N\rangle \frac{1 - 1}{1 - e^{-2\pi ik/r}} \\
&= |1\rangle
\end{aligned}$$

□

Indeed, this property is incredibly useful because we can provide $|1\rangle$ to the lower register of the QPE circuit instead of a single eigenvector, in order to compute the same algorithm but to all the eigenvectors **simultaneously**. We observe that $|1\rangle$ is essentially a superposition of all the eigenvectors of U_x, which means that when fed to C-$U_x^{2^k}$ as target qubit we don't really get $|1\rangle$ back, indeed

$$\begin{aligned}
\text{C-}U_x^{2^k}(|+\rangle_0 \otimes |1\rangle_1) &= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes U_x^{2^k}|1\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes U_x^{2^k}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}U_x^{2^k}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes |1\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{2\pi i 2^k s/r}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2}}\left(|0\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle_1 + |1\rangle_0 \otimes \frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{2\pi i 2^k s/r}|u_s\rangle_1\right) \\
&= \frac{1}{\sqrt{2r}}\sum_{s=0}^{r-1}\left(|0\rangle_0 + e^{2\pi i 2^k s/r}|1\rangle_0\right) \otimes |u_s\rangle_1
\end{aligned}$$

Nevetheless, notice what happened here: we engangled the control qubit with the target qubit, such that now the target *picked up the phase* $e^{2\pi i 2^k s/r}$ — exactly as the standard QPE — and we are still preserving the superposition of eigenvectors $|u_s\rangle$. This is exactly what we need, however it also means that the result at the end of the circuit is not very straightforward: in standard QPE the resulting state is just the tensor product of the single control qubits because they are independent from each other, but here we are entangling all the control and the target qubits together at each application of C-$U_x^{2^k}$. Hence, we need to compute what happens step by step:

fallo

$$\begin{aligned}
&TODO \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 2^{m-1}s/r}|1\rangle\right) \otimes |u_s\rangle \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{2^t}}\bigotimes_{m=1}^{t}\left(|0\rangle + e^{2\pi i 0.\varphi_{s_m}\cdots\varphi_{s_t}}|1\rangle\right) \otimes |u_s\rangle \\
=&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\text{QFT}_t|\varphi_s\rangle_{q_0\cdots q_{t-1}} \otimes |u_s\rangle \\
\xrightarrow{\text{QFT}_t^\dagger(|q_0\cdots q_{t-1}\rangle)}&\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|\varphi_s\rangle \otimes |u_s\rangle
\end{aligned}$$

assuming that $\varphi_s = s/r$ can be written through t bits. This entangled quantum state is exactly what we want, because now by measuring the first t registers we get that for each $s \in [0, r-1]$ they will collapse into $|\varphi_s\rangle$ with probability $\frac{1}{r}$, and the last register

will collapse into $|u - s\rangle$ which is exactly the eigenvector associated to $e^{2\pi i\varphi_s}$. This proves that we truly computed the phases of all the possible eigenvectors simultaneously.

We are almost done. We now know that we can retrieve $\varphi_s = s/r$ for each $s \in [0, r-1]$ without even knowing a single eigenvector of U_x, but there is a problem. Say that we want to find the order of $x = 3$ modulo $N = 11$, and suppose that the true phase is

$$\varphi_s = \frac{s}{r} = \frac{2}{5} = 0.4$$

The QPE algorithm cannot output this exact value, because 0.4 cannot be written precisely in binary — in particular, the reason is that 5 does not divide 2^t for any t. This means that our algorithm will return an *approximation* of φ_s. Say that we are working with 6 registers of precision and fix $t = 6$; then QPE will output the closest approximation of 0.4 with 6 bits:

$$\tilde{\varphi}_s = 0.40625 = \frac{13}{32}$$

Now, it would be a mistake to assume that $r = 32$! How do we proceed?

Definition 4.6: Continued fraction

Given a rational number $x \in \mathbb{Q}$, the continued fraction of x is composed by a set of numbers $a_0, \ldots, a_n \in \mathbb{N}$ — where $a_n \neq 0$ — such that

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots + \cfrac{1}{a_n}}}}$$

The continued fraction is usually denoted with the following notation

$$x = [a_0; a_1, a_2, \ldots, a_n]$$

Moreover, each "partial" continued fraction

$$\begin{aligned}
&[a_0] \\
&[a_0; a_1] \\
&[a_0; a_1, a_2] \\
&\ldots \\
&[a_0; a_1, a_1, \ldots, a_n]
\end{aligned}$$

is called **convergent** of the original number.

For instance, the number $\frac{338}{121}$ can be written as follows:

$$\frac{338}{121} = [2; 1, 3, 1, 5, 4]$$

and its convergents are

$$[2]$$
$$[2;1]$$
$$[2;1,3]$$
$$\dots$$
$$[2;1,3,\dots,4]$$

We will use continued fractions to recover r from s/r. Suppose that QPE outputs $\tilde{\varphi}_s = \frac{13}{32}$ as in the previous example. Without going into the details, there is a classical algorithm which is able to recover its continued fraction in $O(L^3)$ time — where $L = \lceil \log N \rceil$. Thus, we get that

$$\tilde{\varphi}_s = 0 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{6}}} = [0;2,2,6]$$

Now, to recover $\frac{2}{5}$ we just need to compute the convergents:

$$[0] = 0$$
$$[0;2] = \frac{1}{2}$$
$$[0;2,2] = \frac{2}{5}$$
$$[0;2,2,6] = \frac{13}{32}$$

Finally, since we know that r is the order of x modulo N, we just need to pick the convergent with the smallest denominator r such that $x^r \equiv 1 \bmod N$. In our example we had $x = 3$ and $N = 11$, thus

$$3^2 \equiv 9 \not\equiv 1 \bmod 11$$
$$3^5 \equiv 243 \equiv 1 \bmod 11$$

This is how we can recover $r = 5$ — to be precise, there are theoretical results which guarantee that $\frac{2}{5}$ shows between the convergents of $\frac{13}{32}$ but this is ouside the scope of our discussion.

4.5 Shor's algorithm

TODO intro

Theorem 4.3: Fundamental Theorem of Arithmetic

Given any $N \in \mathbb{N}$, there exists unique $p_1, \dots, p_m \in \mathbb{P}$ primes and $\alpha_1, \dots, \alpha_m$ exponents — for some $m \in \mathbb{N}$ — such that

$$N = p_1^{\alpha_1} \cdot \ldots \cdot p_m^{\alpha_m}$$

The set of primes $p_1, \ldots, p_m \in \mathbb{P}$ is called **prime factorization** of N. The **Integer Factorization Problem (IFP)** asks to retrieve the prime factorization of any given $N \in \mathbb{N}$.

Theorem 4.4

Given any L-bit long composite natural number $N \in \mathbb{N}$, let $x \in [2, N]$ different from $N-1$ be a solution to the equation

$$x^2 \equiv 1 \bmod N$$

Then, at least one of $\gcd(x-1, N)$ and $\gcd(x+1, N)$ is a non-trivial factor of N that can be computed using $O(L^3)$ operations.

Theorem 4.5

Given any $N \in \mathbb{N}$ odd composite positive integer with m prime factors, let $x \in_R [1, N-1]$ such that $\gcd(1, N-1) = 1$. Then, if r is the order of x modulo N, it holds that

$$\Pr[r \text{ is even and } x^{r/2} \not\equiv 1 \bmod N] \geq 1 - \frac{1}{2^m}$$

TODO

TODO TODO

TODO

write the alg and explain stuff?

size of QPE

write QPE alg

size of Ux gates

5

Quantum Key Distribution

Why is Shor's algorithm so important? Most widely used **public-key cryptography** algorithms rely on the difficulty of one of the three following mathematical problems:

- the Integer Factorization problem
- the Discrete Logarithm problem
- the Elliptic-Curve Discrete Logarithm problem

In particular, all of the problems are currently *vary hard* to solve efficiently, and basically any modern cryptosystem is based on the fact that we don't know any fast-enough algorithm that is able to solve them. However, all of these could be easily solved on a sufficiently powerful quantum computer that runs Shor's algorithm, as we already seen.

explain details

Nevertheless, at the time of writing quantum computers lack the processing power to break widely used cyrptographic algorithm, so we are still safe from a world without crypto. However, because of the length of time required for migration to **quantum-safe cryptograpy**, are already desining new algorithms to prepare for the so called **Q-Day**, the day when current algorithms will be vulnerable to quantum computing attacks. For example, NIST already initiated a proces to quotes "solicit, evaluate and standardize one or more quantum-resistand public-key cryptography algorithms".

In this chapter we will cover

vedi de che parla prima de parla a vuoto

is a **Quantum Key Distribution (QKD)** protocol. In particular, we are interested in finding secure protocols that able to allow two parties to share a *private key* quantum-safely.

The protocol that we will present is usually called BB84, and it was published in 1984 by Bennett and Brassard [BB14]. Suppose that our two usual protagonists Alice and Bob need to share a key quantum-safely. Alice will choose two random binary strings $(4+\delta)n$-long which are usually referred to as a and b. Then, depending on the bits of these two strings she will construct a superposition of states based on the following table:

a_i	b_i	Chosen state
0	0	$\|0\rangle$
1	0	$\|1\rangle$
0	1	$\|+\rangle$
1	1	$\|-\rangle$

For example, if her two random strings are

$$a = 01101$$

$$b = 00110$$

then she will construct the following superposition

$$|\psi\rangle = |0\rangle + |1\rangle + |-\rangle + |+\rangle + |1\rangle$$

Now, we will assume that Alice and Bob share a communication channel that can be eavesdropped by a third party Eve that wants to know their shared key. Alice will send $|\psi\rangle$ to Bob through this medium. Before describing what happens when Eve comes into play, we will describe the complete protocol.

Let's do a step back: each time we talked about "measuring" we never actually mentioned that when we are *physically* performing a measure what we are really performing is applying a "polarizing lens" to the final state. For instance, this operation can be implemented through Faraday rotators, which leverage the **Faraday effect**:

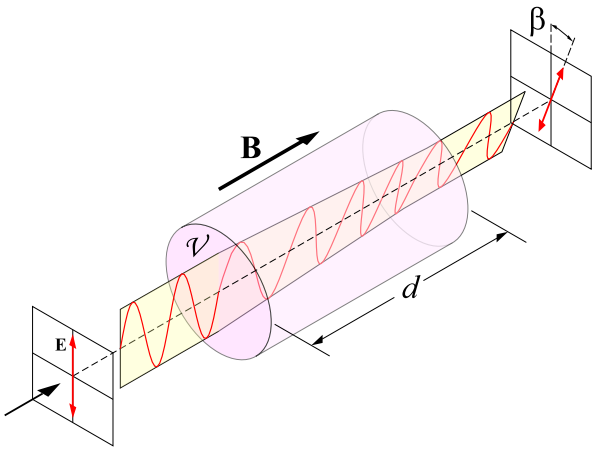


Figure 5.1: A Faraday polarization rotator.

This is important because it means that when we talk about measurement we need to define the "angle" at which we perform it. Mathematically speaking, this is equivalent to defining the "base of choice" for the measurement — since we always consider orthonormal basis only. The importance of the choice of the basis can be illustrated with the BB84 protocol.

- If we have a qubit in the state $|\phi\rangle = |+\rangle$, we know that the probability of getting either $|0\rangle$ or $|1\rangle$ when measured in the canonical basis — formed by the vector $|0\rangle$ and $|1\rangle$ themselves — is precisely 50% for both outcomes because

$$\Pr[\text{measure}(|\phi\rangle = |0\rangle)] = |\langle 0|+\rangle|^2 = \frac{1}{2}$$

- Now, consider the orthonormal basis formed by the vector $|+\rangle$ and $|-\rangle$ — which we will call the **diagonal basis**; we observe that this space is just a rotation of 45 clockwise. This space is interesting: when we measure a qubit $|\phi\rangle$ set to the state $|0\rangle$ in the canonical base we get $|0\rangle$ with 100% certainty, but in this space we have that

$$\begin{aligned}\Pr[\text{measure}(|\phi\rangle = |+\rangle)] &= |\langle 0|+\rangle|^2 \\ &= \left|\frac{1}{\sqrt{2}}\langle 0|\left(|0\rangle + |1\rangle\right)\right|^2 \\ &= \left|\frac{1}{\sqrt{2}}(1+0)\right|^2 \\ &= \frac{1}{2}\end{aligned}$$

  Indeed, as we would expect $|0\rangle$ and $|1\rangle$ in the diagonal basis behave exactly as $|+\rangle$ and $|-\rangle$ do in the canonical one.

We will see how the BB84 protocol takes advantage of this fact later in our discussion.

Now that Alice has sent her superposition $|\psi\rangle$, she will send it to Bob which will receive $\mathcal{E}(|\psi\rangle)$, where $\mathcal{E}$ describes the quantum operation due to the combined effect of the channel and Eve's actions. If Bob correctly receives the information, he publicly announces this fact.

Bob will then proceed to generate a random string of $(4+\delta)n$ bits as well, which we will call b', and based on this very string he will measure the received quantum superposition — in particular, if $b'_i = 0$ he will use the canonical base, and if $b'_i = 1$ he will employ the diagonal one. In our example, assuming that Bob generated the random string

$$b' = 10111$$

he will measure something like

$$\text{measure}(|\psi\rangle) = |-\rangle + |1\rangle + |-\rangle + |+\rangle + |+\rangle$$

We observe that the first and the last bit randomly collapsed to $|-\rangle$ and $|+\rangle$ respectively because they were measured in the wrong basis — indeed, the original string was $b = 00110$ and $b \oplus b' = 10001$. Bob then tries to reconstruct the original string a based on its b', thus getting the string

$$a' = 11100$$

We observe that Bob this string of no use for now, and this is important when we will discuss Eve's role in the protocol.

After Alice has heard that Bob received the state, she can proceed to send b itself as it is over a public channel, and Bob can then discard all the bits of a' that were generated through wrong bits of b' — we observe that their remaining bits satisfy $a = a'$. This step is usually called **basis reconciliation**, and the expected number of qubits kept by Bob after this step is

$$\mathbb{E}[\text{kept qubits}] = \sum_{i=1}^{(4+\delta)n} 1 \cdot \Pr[b_i = b_i'] = \sum_{i=1}^{(4+\delta)n} \frac{1}{2} = \frac{1}{2} \cdot (4+\delta)n = \left(2 + \frac{\delta}{2}\right) n$$

We will assume that both Alice and Bob will just weep $2n$ bits of their result — δ can be chosen sufficiently large so that this can be done with exponentially high probability.

TODO

what happens if eve looks

Bibliography

[BB14] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Theoretical Computer Science* 560 (Dec. 2014), 7–11. ISSN: 0304-3975. DOI: 10.1016/j.tcs.2014.05.025. URL: http://dx.doi.org/10.1016/j.tcs.2014.05.025.

[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.

[Gro05] Lov K. Grover. "Fixed-Point Quantum Search". In: *Physical Review Letters* 95.15 (Oct. 2005). ISSN: 1079-7114. DOI: 10.1103/physrevlett.95.150501. URL: http://dx.doi.org/10.1103/PhysRevLett.95.150501.

[Gro97] Lov K. Grover. "Quantum Mechanics Helps in Searching for a Needle in a Haystack". In: *Physical Review Letters* 79.2 (July 1997), 325–328. ISSN: 1079-7114. DOI: 10.1103/physrevlett.79.325. URL: http://dx.doi.org/10.1103/physrevlett.79.325.

[Kit95] A. Yu. Kitaev. *Quantum measurements and the Abelian Stabilizer Problem*. 1995. DOI: 10.48550/ARXIV.QUANT-PH/9511026. URL: https://arxiv.org/abs/quant-ph/9511026.

[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.

[Sho] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. SFCS-94. IEEE Comput. Soc. Press, 124–134. DOI: 10.1109/sfcs.1994.365700. URL: http://dx.doi.org/10.1109/SFCS.1994.365700.

[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.

[YLC14] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang. "Fixed-Point Quantum Search with an Optimal Number of Queries". In: *Physical Review Letters* 113.21 (Nov. 2014). ISSN: 1079-7114. DOI: 10.1103/physrevlett.113.210501. URL: http://dx.doi.org/10.1103/PhysRevLett.113.210501.