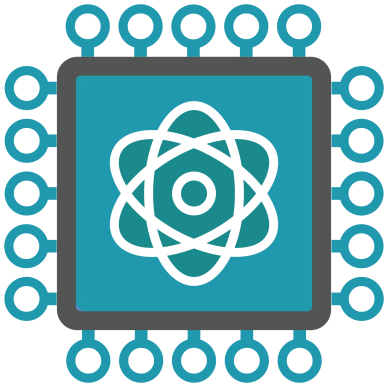
TÉCNICO LISBOA



# A Quantum Leap Into the Future of Computer Science

**Paulo Tiago Gouveia Saldanha**

Thesis to obtain the Master of Science Degree in

## Electrical and Computer Engineering

Supervisors: Prof. Doutor Carlos Manuel dos Reis Paiva
Prof.ª Doutora Filipa Isabel Rodrigues Prudêncio

### Examination Committee

Chairperson: Prof. Doutor José Eduardo Charters Ribeiro da Cunha Sanguino
Supervisor: Prof. Doutor Carlos Manuel dos Reis Paiva
Member of the Committee: Prof. Doutor Marco Alexandre dos Santos Ribeiro

**November 2022**

*To the memory of my father*,

*Paulo da Conceição Rodrigues Saldanha*

*SIC PARVIS MAGNA*

*- Vergil, Eclogue I.19–23*

# Acknowledgments

Em primeiro lugar gostaria de dar o meu agradecimento ao Professor Doutor **Carlos Paiva** pela oportunidade e honra de poder explorar esta área sob sua tutoria. Em segundo lugar, à Professora Doutora **Filipa Prudêncio** pela sua ajuda e orientação que permitiram o desenvolvimento desta dissertação.

Destaco as quatro pessoas mais importantes da minha vida que são os maiores responsáveis pela minha formação pessoal e académica. Obrigado por me tornarem no filho, irmão e neto mais priveligiado do mundo.

Quero agradecer ao meu saudoso pai, **Paulo da Conceição Rodrigues Saldanha**, por tudo o que me deu e ensinou. Por todos os momentos que passámos juntos e que me formaram enquanto pessoa e estudante. Obrigado pela jornada que partilhámos. Este trabalho além de ser dedicado a si, é também seu. ”És o meu herói e serás sempre o homem que escolhi para meu melhor amigo”.

À minha mãe, **Maria Celina Delgado de Gouveia Saldanha**, não só pelo seu cuidado e carinho mas também pelo seu esforço, dedicação e persistência que foram sem dúvida dos maiores pilares da minha formação académica. Foste e és excecional. Isto também é um trabalho teu, mãe. Parabéns por isso.

À minha irmã, **Sara Raquel Gouveia Saldanha** por ser a minha fiel parceira desde sempre e por me tornar num bom exemplo. O impacto que tiveste para a minha formação é algo de inédito.

À minha avó **Maria** pela sua cumplicidade, ensinamentos e apoio, que foram muito importantes na minha educação. Obrigado por alimentar as minhas forças desde sempre.

Também dedico este trabalho a toda a minha restante família (tios, tias, primos, primas e respetivos cônjugues): às minhas tias Zé, Paz, Luísa, Helda, Verónica e Lúcia; aos meus tios João Luís, George, Luís Alves e Ricardo; aos meus primos e primas Telma, Cristina, Cynthia, Bárbara, André, Georgio, Liana, Ana Margarida, João, Isabel, Francisco, Elisabete, Miguel, Alexandre, Sandra, Márcio, Chantell, Sérgio, Gina (e filhas), Bianca, Juan-Pierre, Mariana, Nicole, Daniel, Miguel Ângelo, Christian, Francisca, Bárbara, Ivy, Chico, Guilherme, Constança, Henrique, Gabriel, Matilde, Lillian, Rodrigo e Simão.

Aos meus amigos de sempre: Marco Ornelas, Vitor Hugo, Élio Gouveia, Marco Abreu, Gonçalo Fernandes, Rúben Ferreira, Tiago Gonçalves, Rui Milho, Joana Figueiredo, Ana Ramalhosa, Artur Nóbrega e Gonçalo Parrinha.

Aos meus prezados Professores, Professoras e pessoal não-docente do Colégio Infante D. Henrique e da Escola da APEL.

A todos os colaboradores do Supermercado Regional e da RC Automação.

Aos meus colegas e treinadores do Clube Desportivo Infante e do Madeira SAD.

Gostaria igualmente de agradecer ao Instituto de Telecomunicações (IT) do Instituto Superior Técnico.

Todos vós fizeram e/ou fazem parte da minha vida, marcaram-na e definiram, de alguma maneira, o sucesso que hoje tenho. Fica o meu Muito Obrigado.

# Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

# Resumo

Nos últimos cem anos, a área da mecânica quântica passou de alvo de críticas e incertezas para uma realidade do nosso quotidiano. O paradigma desta teoria mudou em 1964 quando John Stewart Bell teorizou que a mecânica quântica era incompatível com qualquer teoria clássica, nomeadamente o realismo local. O teorema de Bell mostrou-se ser um dos maiores avanços na física quântica pelo que rapidamente começou a ser aplicado em várias áreas da nossa sociedade, incluindo computação.

A computação quântica é uma área baseada em princípios da física quântica, tendo como objetivo solucionar problemas demasiado complexos para computadores clássicos. O potencial desta área é enorme pelo que investigadores de todo o mundo estão dedicados a explorar todas as suas vantagens.

Esta dissertação é centrada em computação quântica e por essa razão, começa por dar definições essenciais para aqueles que não estão familiarizados com a mecânica quântica (mas que têm bases de álgebra linear). Alguns conceitos como *bits* quânticos, entrelaçamento quântico e sobreposição são primeiramente introduzidos para que definições *a posteriori* sejam de fácil compreensão. Seguidamente, é explorado o tema de computação quântica onde noções como *gates* quânticas, teorema *no-cloning*, *superdense coding*, teletransporte quântica, protocolos criptográficos, correção de erros e o algoritmo de Simon serão analisados. Finalmente, será feita uma breve introdução à teoria das categorias apresentando conceitos como categoria, objeto, morfismo, functor e transformações naturais de maneira que se possa seguidamente analisar a nova forma de se representar (através de diagramas) computação quântica e informação quântica: pictorialismo quântico.

**Palavras-chave:** quântico, entrelaçado, qubit, algoritmo, categoria.

# Abstract

For the last one hundred years, the field of quantum mechanics came from a target of criticism and doubt to an everyday reality. The paradigm of this theory changed in 1964 when John Stewart Bell showed that quantum mechanics was incompatible with any classical theory, including local realism. Bell's theorem was one of the biggest breakthroughs in quantum physics for which started to get applications in many fields of our society, including computer science.

Quantum computation is an area based on the principles of quantum physics, focused to solve problems that are too complex for classical computers. The potential in this area is astonishing, for which researchers around the globe are very committed to exploit all its advantages.

This dissertation is centered on quantum computing and for that reason it starts by giving some essential tools for those who are not so familiar with quantum mechanics (but with a previous background on linear algebra). Some basic concepts like quantum bits, quantum entanglement and quantum superposition are early introduced to better understand *a posteriori* definitions. After that, the field of quantum computation is exploited, where notions like quantum gates, no-cloning theorem, superdense coding, quantum teleportation, cryptographic protocols, error corrections and Simon's quantum algorithm will be looked in depth. Finally, a brief overview on category theory is done by giving some essential concepts namely, category, object, morphism, functor and natural transformations, in order to analyze a novel way to present (trough diagrams) quantum computation and quantum information: quantum picturalism.



**Keywords:** quantum, entanglement, qubit, algorithm, category.

# Contents

# List of Tables

# List of Figures

# Acronyms

**BB84** Bennett-Brassard-1984

**CHSH** Clauser–Horne–Shimony-Holt

**CNOT** Controlled-NOT

**EPR** Einstein–Podolsky–Rosen

**PR** Popescu-Rohrlich

**RSA** Rivest–Shamir–Adleman

# Chapter 1

# Introduction

## 1.1 State-of-the-Art

The constant improvement of technology allowed mankind to build the most powerful machines that revolutionized the world. Since the last century, the computer science area is making huge developments, creating smaller and more powerful computers every day to aid our quotidian needs. Although classical computers can do amazing things, they struggle to solve certain kinds of problems by either not presenting a solution or partially solve small versions of the problem. This means that if the problem is big enough, normal computers run out of computing power and get stagnated to perform some tasks, precluding new and great technological advances.

In 1965 Gordon Moore (1929-) made an observation, the so-called Moore's Law, which stated that the number of transistors in an integrated circuit doubled every two years. This law held true for the last decades, but twenty years into the twenty-first century, computer parts are reaching their physical limit.

In order to solve these problems, for the last forty years scientists combined two of the most influential and revolutionary theories: information theory and quantum mechanics. This combination allowed new approaches for solving complex problems in fundamentally new ways, by using quantum physics to perform computations.

The effects of quantum superposition and entanglement of qubits allow quantum computers to do things that are absolutely impossible with classical computers. They are faster, do better simulations and estimations and even act better in the many fields of our society. The world as we know it can change by achieving full power of quantum computing. For example, incurable diseases today have the potential to be fully treatable. And that is not all. It can be easier to diagnose those diseases and to create personalized medicine to help a specific person with a specific problem. Our privacy will also be improved and researching fields such as Artificial Intelligence and Machine Learning can make unimaginable advances [1].

Nevertheless, quantum mechanics used to have an infamous reputation and is still hard to understand. One should ask, even before presenting quantum computing, why do we struggle so much to understand quantum mechanics? Richard Feynman (1918-1988), a Nobel Prize in Physics (1965), once

said “If you think you understand quantum mechanics, you do not understand quantum mechanics” [2]. Sir Roger Penrose (1931-), another Nobel Prize in Physics (2020) as too stated “Quantum mechanics makes absolutely no sense” [3]. If these Nobel Prize winners present these arguments upon quantum mechanics, what is the hope for us? As we will expose during the course of this dissertation, quantum mechanics is a very counter-intuitive science due to the fact that the consequence of its laws cannot be seen as the consequences of classical laws like an apple falling off a tree. We can go even beyond that and ask what is the source problem that bothers even the most prestigious scientists and searchers like no other theory? The answer to this question related with the way that quantum mechanics appeared experimentally. In the 1800's light was interpreted as waves in the electromagnetic field. This phenomenon was described by James Clerk Maxwell (1831-1879) throughout his famous equations that showed us how the electric field and the magnetic field can influence one another [4]. However, experiments like Albert Einstein's (1879-1955) (Nobel Prize in Physics in 1921)[5] photoelectric effect or Arthur Compton's (1892-1962) (Nobel Prize in Physics in 1927)[6] Compton effect demonstrate that light can be described as particles. Something that is simultaneously a wave and a particle is weird when you think about it, but the truth is that light can be interpreted as such and, with that, the concept of wave-particle duality is born. To better understand the weirdness around this topic let us describe an experiment[7, Section 12.III].

The experiment consists in a beam of light and three polarized lenses/disks (for example polarized sunglasses). Polarized lenses can be interpreted as quantum measurement devices since they work as polarizing filters. By using polarizing filters, when a photon reaches the lenses, it either passes through or not, depending on the direction that that photon is polarized. This experiment reveals properties from not only the polarized lenses but also from photons themselves.

Imagine a beam of light coming out of its source. This light is not polarized, meaning that the beam of light is polarized in all directions at the same time, i.e., the electric fields are in all directions. Then we put a polarizer, more concretely, a linear polarizer in the path of the unpolarized light. Suppose the light that gets through is horizontally polarized by the filter (as exemplified in Figure 1.1), implying that all the vertically polarized parts will be absorbed. This means that half the light or half the photons pass through the filter.

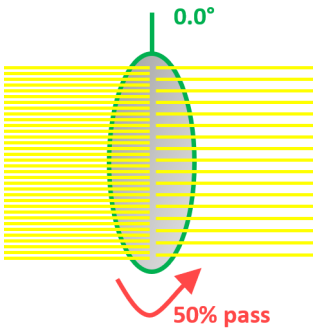



Figure 1.1: Beam of light being emitted through a polarized filter

If a second polarized lens is placed after the first lens, what amount of light should pass through the

second filter? A reasonable guess would be it reduces the intensity of the light again by half. However, that is not the case. Surprisingly, the amount of photons that pass through both disks depends on the orientation of the second disk with respect to the first.

This means that if we rotate the second filter, the lamp will look lighter or darker. When the second lens has the same orientation as the first lens, that is, when the second filter is oriented $0°$ off from the first, every photon that gets through the first lens also gets through the second one. However, when we rotate the second filter, the lamp gets continuously more and more dark until reaches $90°$. When the angle between the first and the second filter is $90°$, the amount of light that passes through the second disk is zero, meaning that all photons are blocked by the second filter when the lenses are perpendicular (see Figure 1.2). If we continue to rotate it, the lamp will eventually get lighter until reaches the $0°$ which is the lightest it can get. Nevertheless, it is when we insert a third filter between the first and the second one that the weirdness of quantum mechanics jumps in.

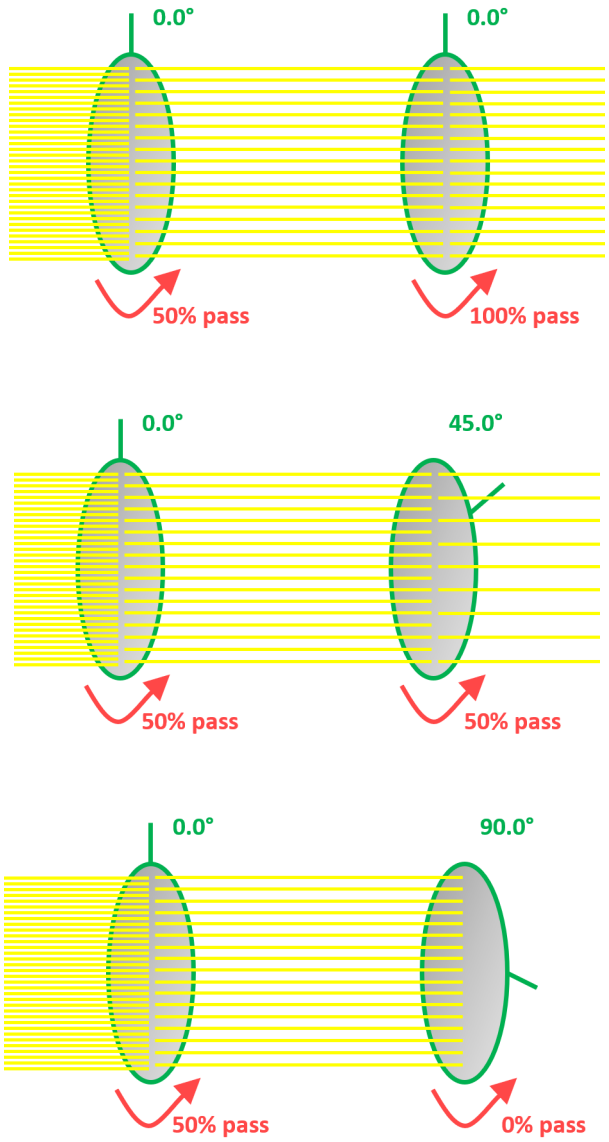



Figure 1.2: Three examples where a beam of light is emitted through two polarized filters with different orientations: $0°$, $45°$ and $90°$

Until this point, both filters have just removed light. But if you take a third filter and orient it $45°$ off from the first filter, that is, put it between the first two filters (sequence of $0°$, $45°$ and $90°$ as illustrated in Figure 1.3), the lamp will actually look brighter. Half the photons from the first lens get through the second lens and half those photons from the second lens get through the third lens since all that matters is the angle between the last two disks as seen before. With that, we easily see that $12.5\%$ of the initial unfiltered light gets through. The $45°$ filter is not generating more photons but is modifying some features of their polarizations. Contrary as one should expect, the more filters between the first ($0°$) and the last filter ($90°$), the brighter the lamp gets. The weirdness here is if two filters obstruct $100\%$ of the photons, how can we get more light by adding more filters (more obstacles) between the first two? This shows that filters are not passive. They are active: they interact with light.

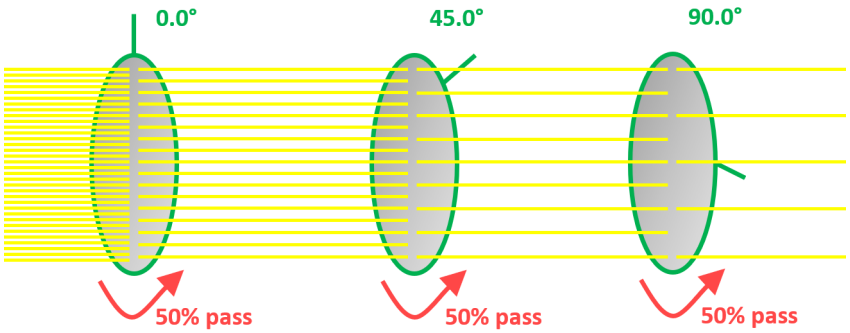



Figure 1.3: Beam of light emitted through three polarized filters

The quest for knowledge was, is and will always be the fuel that keeps Humanity moving forward into a more practical, sustainable, healthier and safer future. The weirdness and the sense of surprise that something unknown can provoke in Humans is the beginning of a search for answers. Nowadays, the most successful tool use to achieve those answers is without a doubt the computer.

There is no denying that the Turing machine (invented in 1936 by Alan Turing (1912-1954)) made a huge impact in the way we live: thanks to its concept, it allowed the expansion of the computer science and artificial intelligence fields. It becomes harder and harder to think how different our lives could be without the existence of something as banal (at least nowadays) as a computer. Computers are (literally) everywhere, on the desktop at home, at work, at the library, even in our pockets and bags. Why? Because facilitates our quotidian needs, from contacting someone to run an algorithm that puts power stations operating.

In the 1980's the idea of using computers to solve quantum problems started to get some attention. In fact, it almost became necessary to get aid from a computer in order to exploit the consequences of Bell's theorem (named after John Stewart Bell (1928-1990)) regarding quantum non-locality. Nevertheless, it was a few years after that the first quantum algorithms started to emerge. Quantum algorithms proved to have massive advantages over classical ones due to their exponential speed-up in processing.

For the last forty years various quantum algorithms were developed, resulting in bigger and more complex quantum circuits. For this reason, the necessity to simplify circuits became eminent. The solution for this optimization was found in the early 2000's by Samson Abramsky (1953-) and Bob Coecke

(1968-) when they merged quantum information with a mathematical abstract language named category theory [8]. The consequence of this combination is one of the main purposes of this dissertation for which it will be deeply analyzed in the closing chapters.

## 1.2 Objective

The main purpose of this work is to acquire knowledge in one of, if not the most, important field of the future of mankind: quantum computation. In fact, coincidentally with the time this dissertation was being composed, the Nobel Prize of Physics 2022 was attributed to three physicists named Alain Aspect (1947-), John Clauser (1942-) and Anton Zeilinger (1945-) for their contributions in quantum information by conducting experiments with entangled photons [9]. Their results have demonstrated that quantum mechanics can have various applications, especially in fields like quantum computing, quantum networks and quantum encrypted communications. Although quantum computers are not replacements of classical computers, they are vastly superior in some areas since they can take advantage of quantum properties.

The dissertation aims to gently introduce quantum mechanics, quantum computing, quantum algorithms and categorical quantum mechanics applied to computation in a clear and comprehensible way, with theoretical explanations always followed by examples (which are always performed with the same volunteers, Alice and Bob) so the reader can smoothly and gradually assimilate the topics discussed. The reason for this resides in the purpose to reach anyone interested in these themes, even if they have never study quantum mechanics. Nevertheless, a small background in linear algebra is highly recommended.

Being a thesis to obtain a Master of Science Degree in Electrical and Computer Engineering and as the title suggests, the main focus of this dissertation will be quantum computation. Despite the fact that quantum computers already exist, they are only owned by top quantum computing companies and research labs such as Microsoft Quantum Computing, IBM Quantum Computing and Google Research [10]. For this reason, this dissertation will be theoretical since its information will be acquired through the available public sources.

## 1.3 Structure of the Document

This dissertation contains 5 Chapters. The first chapter, the current one, introduces the work of the dissertation by giving a historical background and a light-experiment to give the reader a taste of the spookiness that comes after. It is also presented the structure of the dissertation and the original contributions.

The second chapter is dedicated to quantum mechanics and quantum computing. We start from the rock-bottom of the field: quantum bits and their representation on the Bloch Sphere; quantum properties such as the quantum entanglement and superposition; a quantum game based on Popescu-Rohrlich (PR) boxes, which highlights the discrepancy of results between the Einstein–Podolsky–Rosen (EPR)

reasoning and quantum mechanics; a brief but essential description on quantum gates, from one input gates to two or more, associated with the quantum oracle and two examples with an application of those gates, a short Walsh-Hadamard Transform example and the notorious Hardy state experiment.

We then enter Chapter 3 which is more focused in the quantum computing theme, namely, quantum circuits and quantum algorithms. We start by presenting three fundamental properties that are recurrent when dealing with quantum circuits: the no-cloning theorem, superdense coding and quantum teleportation. Next it is presented the BB84 protocol, the first quantum cryptographic protocol used to improve safe communications. Then we address quantum error correction: what is it, why it happens and how correct it in order to achieve reliable and noiseless communications. The chapter ends with an explanation of Simon's algorithm, a quantum algorithm that emphasizes the exponential difference between classical and quantum algorithms.

In Chapter 4 we make use of all the topics exposed in previous chapters so we can represent them in a new formalism: categorical quantum mechanics. The chapter is chronologically organized, it firstly goes to the beginnings of set theory, what is it and what is its place in mathematics, then we dive into category theory, explaining how it differs from set theory, introducing some properties of the theory like identity, associativity, morphism, product, coproduct, functors and natural transformations. The chapter wraps-up with the main topic of the dissertation: quantum picturalism. This last part is dedicated to explore the new formalism pioneered by Bob Coecke where we first present some simple diagrams and properties; then we show how is possible to represent quantum states and quantum processes; next it is given a demonstration of the no-cloning theorem, but this time diagrammatically; it is introduced the ZX-calculus, a new graphical language used to manipulate quantum maps and some of its rules and representations; we end with a demonstration of a quantum oracle drawn with this new formalism.

The fifth and final chapter regards some final comments from each chapter as well as prospects of future work.

## 1.4 Original Contributions

This dissertation was solely built from the public sources, namely: books, articles and online lectures. The original contributions of this work includes images, diagrams and tables since they were developed by the dissertation's author. Nevertheless, one should note that part of the categorical quantum diagrams in Chapter 4 are based on the content of the book 'Picturing Quantum Processes: A first course in quantum theory and diagrammatic reasoning'[11].

Although the issues covered in most part of this work are dated for more than seventy years it is when we dive into categorical quantum mechanics that we enter uncharted territory. This new formalism was proposed to be in this Master's dissertation for being a quite recent topic and a promising innovative way to aboard the quantum computing area.

# Chapter 2

# Quantum Computation: An Overview

In the beginnings of the last century, Max Planck's (1858-1947) work in thermodynamics led to the birth of quantum mechanics [12], a fundamental theory that explains natural phenomena at the scale of atoms and subatomic particles. Scientists tried for decades to understand and develop technologies that could rely upon quantum effects, but it was in 1982 that Richard Feynman published the idea that it should be possible and necessary to build a quantum computer [13]. The firsts realizations of this idea were made by a British physicist named David Deutsch (1953-), one of the pioneers of quantum computation. His work showed that a quantum computer has an incredible advantage over a classic computer by doing things that a classic computer cannot [14].

But what exactly are the differences between a classic computer and a quantum computer?

## 2.1 Quantum Bits

A classical computer has computer chips. Each chip incorporates modules which contains logic gates. These gates perform operations using classical bits which can be either one of two states: $0$ or $1$. The combination or a string of several bits is used to represent and process information.

On the other hand, quantum computers use quantum bits or *qubits* [15, Sections 1.2 and 1.3]. These qubits are represented by what Paul Dirac (1902-1984) called a ket[1]. Qubits are formally represented as states such as $|0\rangle$ and $|1\rangle$ which are often referred as the computational basis.

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.1}$$

In a quantum computer if we have two qubits, we get a Hilbert state[2] of dimension $2^2 = 4$

$$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}, \tag{2.2}$$

[1] In quantum mechanics, the bra–ket notation is used to represent quantum states where $\langle \ | \ \text{is a bra and} \ | \ \rangle$ is a ket. One should note that a ket is a unit vector. To prove that, we just need to multiply a ket for its respective bra, e.g., $\langle \psi \mid \psi \rangle = \begin{pmatrix} \alpha_0^* & \alpha_1^* \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = |\alpha_0|^2 + |\alpha_1|^2 = 1$

[2] A Hilbert space is the state of a physical system represented by a complex vector space with an inner product. It was firstly introduced in 1932 in John von Neumann's (1903-1957) book 'Mathematical Foundations of Quantum Mechanics'

and for three qubits we have a Hilbert state of dimension $2^3 = 8$.

$$\{|000\rangle, |001\rangle, |010\rangle, |011\rangle, |100\rangle, |101\rangle, |110\rangle, |111\rangle\} \tag{2.3}$$

If it were $100$ qubits, we would have $2^{100} \approx 1.268 \times 10^{31}$ states. But is it even possible to a computer to manipulate such an astronomical number of states? Well, for sake of simplicity let us use smaller numbers like $N = 6$. In this case, we have $2^6 = 64$ states and we want to find one specific state. A normal computer could take thousands of steps to find it because it can only be in one basic state at a time. However, a quantum computer could only take four steps to find it due to one very important property called superposition. When a quantum computer is in a superposition, it can process many quantum states in parallel, making it exponentially faster to find the intended state.

A generic qubit $|\psi\rangle$ is a superposition or linear combination as described in (2.4),

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{2.4}$$

where $\alpha$ and $\beta$ are complex numbers. Note that $|\alpha|^2 + |\beta|^2 = 1$ since $\alpha^2$ represents the probability of the qubit being in the state $|0\rangle$ and $\beta^2$ the probability of the qubit being in the state $|1\rangle$. With this result, we can affirm that a qubit's state is a unit vector in a two-dimensional complex vector space.

The ability of a qubit to be in a superposition state means that it is in a continuum of states between $|0\rangle$ and $|1\rangle$. In other words, if a qubit is not measured, is as if it was in a limbo state of $0$ and $1$, simultaneously. But when measured, the qubit will collapse into state $0$ with some probability or into state $1$ with another probability, depending on the values of $\alpha$ and $\beta$. For example if $\alpha = \sqrt{\frac{1}{2}}$ and $\beta = \sqrt{\frac{1}{2}}$ we have

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \tag{2.5}$$

which means that the qubit has a $|\alpha|^2 = \left|\frac{1}{\sqrt{2}}\right|^2 = 0.5 = 50\%$ chance of being at state $0$ and a $|\beta|^2 = 50\%$ chance of being at state $1$.

## 2.2 Bloch Sphere

In general, it is useful to use geometric representation when using qubits. For that, let us take into account two real positive numbers: $r_0$ and $r_1$,

$$r_0, r_1 \in \mathbb{R}_0^+ = \{x \in \mathbb{R} \mid x \geq 0\} \tag{2.6}$$

and two phases, $\varphi_0$ and $\varphi_1$,

$$\varphi_0, \varphi_1 \in [0, 2\pi] \tag{2.7}$$

such that the previous amplitudes $\alpha$ and $\beta$ can be written as

$$\begin{aligned} \alpha &= r_0 e^{i\varphi_0} \in \mathbb{C}, \\ \beta &= r_1 e^{i\varphi_1} \in \mathbb{C}. \end{aligned} \tag{2.8}$$

With this new notation, expression (2.4) can be re-written as

$$|\psi\rangle = e^{i\varphi_0} \left[ r_0 |0\rangle + r_1 e^{i(\varphi_1 - \varphi_0)} |1\rangle \right] \tag{2.9}$$

but since the global phase $\varphi_0$ affects equally both states, it does not have any physical relevance meaning that

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{2.10}$$

where $r_0 = \cos\left(\frac{\theta}{2}\right)$, $r_1 = \sin\left(\frac{\theta}{2}\right)$ and $\phi = \varphi_1 - \varphi_0$.

Given expression (2.10), it is possible to represent geometrically a qubit $|\psi\rangle$ and its operations applying spherical coordinates ($\theta$ for latitude and $\phi$ for longitude) by using a unit sphere called *Bloch sphere* (illustrated in Figure 2.1) [16, Section 2.4].

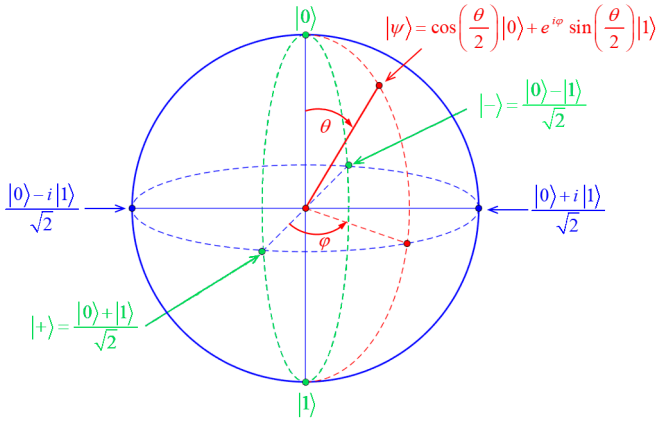



Figure 2.1: Bloch sphere representation of a qubit

As we can easily verify and if we use photons instead, using this basis, the north pole represents a vertical polarization $|0\rangle$ and the south pole represents a horizontal polarization $|1\rangle$. In the equator we have two important points: the $|+\rangle$ which represents a linear polarization at 45° and its antipodal $|-\rangle$, representing a linear polarization at 135°. Note that along the green meridian, we just have linear polarizations ($\varphi = 0$). On the other hand, in the equator we have denoted two other points which corresponds to the right-handed circular polarization and the left-handed circular polarization. Any point in the Bloch sphere that is not on the meridians mentioned are considered states with elliptical polarization.

## 2.3 Quantum Entanglement

Another non-intuitive property that qubits can have is entanglement [17, Section 3.2]. Quantum entanglement happens when for example two quantum particles interact with one another, making their quantum states interdependent. In other words, it starts to happen a correlation between two particles in such a way that when one enters a certain state, the other will react instantaneously in some way, no matter how far apart they are. This phenomenon is called *quantum non-locality*. These particles can now be consider one single quantum entity. Any attempt to decompose this entity in a tensor product of independent quantum states fails. To prove it, let us take into account one of the notorious Bell's state

$$\left|\Phi^{+}\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{2.11}$$

and let us suppose that this state results from a tensor product of two single-qubits (i.e., we are going to prove it by contradiction)

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{2.12}$$

If we develop the left side of the equation we get

$$(a_1|0\rangle + b_1|1\rangle) \otimes (a_2|0\rangle + b_2|1\rangle) = a_1a_2|00\rangle + a_1b_2|01\rangle + b_1a_2|10\rangle + b_1b_2|11\rangle, \tag{2.13}$$

which results in the addition of four terms.

Note that the right side of the expression (2.12) only has $|00\rangle$ and $|11\rangle$, therefore, the terms in expression (2.13) with $a_1b_2$ and $b_1a_2$ must be equal to $0$. However, if $a_1b_2 = 0$ means that either $a_1$ or $b_2$ are $0$ and that cannot happen since $a_1$ multiplies $|00\rangle$ and $b_2$ multiplies $|11\rangle$ in (2.13).

$$(a_1b_2 = 0) \Rightarrow (a_1a_2 = 0) \text{ or } (b_1b_2 = 0) \tag{2.14}$$

The same applies for $b_1$ and $a_2$. If $b_1a_2 = 0$, either $b_1$ or $a_2$ are $0$ and that cannot happen since $b_1$ multiplies $|11\rangle$ and $a_2$ multiplies $|00\rangle$ in (2.13).

$$(b_1a_2 = 0) \Rightarrow (a_1a_2 = 0) \text{ or } (b_1b_2 = 0) \tag{2.15}$$

We have here two contradictions. For this reason, it is possible to conclude that we are indeed in the presence of an entangled state.

Quantum entanglement is one essential property that changed the whole paradigm for quantum computation. If we have entangled qubits, by measuring one qubit, we can directly deduce properties of its partners without having to measure them. This makes the amount of information contained in an entangled state of $N$ qubits grow exponentially instead of linearly as normal computers do [18, Chapter 1].

## 2.4 Quantum Gates

Analogous to classical computers that use logic gates to manipulate bits, quantum computers use quantum gates [15][19, Section 1.3][20, Section 7.6] to carry and manipulate quantum information, changing the system's state. These gates are what move the state vector around the Bloch sphere. Mathematically, quantum gates are represented by a unitary matrix. This section presents the most commonly used quantum gates.

### 2.4.1 Pauli Gates

The Pauli matrices were introduced by the physicist Wolfgang Pauli (1900-1958) in order to study the behavior of spins in electrons. In quantum computation, the Pauli matrices $X$, $Y$ and $Z$ are $2 \times 2$ complex matrices that correspond to operations in certain quantum logic gates.

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \qquad \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.16}$$

For example, while matrix $X$ corresponds to quantum gate NOT[3], the matrix $Z$ is used to introduce a phase of $180°$ between the first and the second bit.

All these matrices have two fundamental characteristics[4]: they are Hermitian and unitary. Note that the definitions of Hermitian and unitary matrix are given by (2.17) and (2.18), respectively.

$$U = U^\dagger \tag{2.17}$$

$$U^\dagger U = U U^\dagger = I \tag{2.18}$$

While a Hermitian matrix is a matrix whose conjugate transpose is equal to itself, an unitary matrix is a matrix whose inverse equals to its conjugate transpose. Therefore, the eigenvalues of a Hermitian matrix are always real and the eigenvalues of a unitary matrix are always complex numbers which are contained in the unitary circle of the complex plane. Therefore, these matrices have real eigenvalues contained in the unit circle, concluding that these values can only be 1 and -1 since they are the only real numbers in the unit circle of the complex plane.

These matrices also have the following properties:

$$X^2 = Y^2 = Z^2 = I \tag{2.19}$$

$$XY = iZ = -YX \qquad ZX = iY = -XZ \qquad YZ = iX = -ZY \tag{2.20}$$

[3] See Appendix A.1 for more information

[4] Although it was not necessary for the following conclusions, there one more characteristic named Projector. See details in Appendix A.2

To describe another interesting propriety about these matrices, let us suppose that

$$u = u_x e_1 + u_y e_2 + u_z e_3, \tag{2.21}$$

$$\sigma = X e_1 + Y e_2 + Z e_3. \tag{2.22}$$

The inner product between $u$ and $\sigma$ will result in

$$u \cdot \sigma = u_x X + u_y Y + u_z Z, \tag{2.23}$$

and if we sum the identity matrix where the diagonal elements are $u_0$, we obtain a complex matrix which corresponds to a linear combination of the Pauli matrices and the identity matrix, i.e., a space basis of $2 \times 2$ complex matrices.

$$M = u_0 I + u \cdot \sigma = \begin{pmatrix} u_0 + u_z & u_x - i u_y \\ u_x + i u_y & u_0 - u_z \end{pmatrix} \tag{2.24}$$

Now, let $n$ be an unitary real vector such that

$$n = n_x e_1 + n_y e_2 + n_z e_3, \tag{2.25}$$

$$n^2 = n \cdot n = n_x^2 + n_y^2 + n_z^2 = 1. \tag{2.26}$$

Given the inner product

$$n \cdot \sigma = n_x X + n_y Y + n_z Z = \begin{pmatrix} n_z & n_x - i n_y \\ n_x + i n_y & -n_z \end{pmatrix} \tag{2.27}$$

Note that $n_x$, $n_y$ and $n_z$ are real numbers. This means that any unitary matrix $2 \times 2$ can be described as one unitary real vector $n$ and a parameter $\theta$, i.e., the unitary matrix defined in (2.27) corresponds to a rotation in the Bloch sphere where $n$ is the rotation axis and $\theta$ the rotation angle. For example let us suppose that we have a point of the Bloch sphere in Figure 2.1 that corresponds to a quantum state $|\psi\rangle$. If we apply a quantum gate (X,Y or Z), $|\psi\rangle$ will evolve to another quantum state $|\phi\rangle$, which corresponds to another point in the Bloch sphere.

### 2.4.2 Hadamard Gate

One very important gate in quantum computation is the *Hadamard* gate named by the French mathematician Jacques Hadamard (1865-1963).

$$H = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \langle 0| + \frac{|0\rangle - |1\rangle}{\sqrt{2}} \langle 1| \tag{2.28}$$

The matrix representation of the Hadamard gate is given by (2.29).

$$H = \frac{1}{\sqrt{2}}(X+Z) = \frac{1}{\sqrt{2}}\left[\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix} + \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\right] = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \tag{2.29}$$

This gate allows to create a superposition given a basis state. For example when Hadamard gate is applied to a $|0\rangle$, it transforms it into a $|+\rangle$ and when applied to a $|1\rangle$ it turns it into a $|-\rangle$.

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \tag{2.30}$$

$$H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle \tag{2.31}$$

The Hadamard gate possess some properties such as

$$H^2 = I \qquad HXH = Z \qquad HZH = X. \tag{2.32}$$

One should note that if we apply a Hadarmad gate to three or more bits, we should make use of expression (2.33),

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1}(-1)^{x\odot y}|y\rangle \tag{2.33}$$

where

$$x \odot y = \sum_{m=0}^{n-1} x_m y_m = x_0y_0 \oplus x_1y_1 \oplus \cdots \oplus x_{n-2}y_{n-2} \oplus x_{n-1}y_{n-1} \tag{2.34}$$

This last equation is a consequence of the Walsh-Hadamard transformation (more of this on the current section). Figure 2.2 depicts a Hadamard gate on quantum computation.


Figure 2.2: Typical representation of Hadamard gate in quantum computation

### 2.4.3 Controlled-NOT Gate

Another commonly used quantum gate is the Controlled-NOT (CNOT) gate $C_{ij}$ where $i$ represents the control bit and $j$ the target bit. This gate has two versions: $C_{10}$ and $C_{01}$.

The version $C_{10}$ is a gate applied to two qubits, $x$ and $y$, and outputs two qubits, $x$ and $y \oplus x$. These qubits can assume the values of either $0$ or $1$.

$$C_{10}|x,y\rangle = |x, y \oplus x\rangle \tag{2.35}$$

$$C_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.36}$$

This gate basically performs the NOT operation on the second qubit only when the first qubit is True, i.e., $1$. For example if the first input qubit is $x = 0$, the second output qubit will be the input $y$, because $y \oplus 0 = y$. However, if the control bit is activated, that is, if $x = 1$, the second output qubit will be the negation of the second input qubit.

For the version $C_{01}$ is precisely the opposite. When the control bit, which is the second input qubit, is $1$, the first output qubit will be the negation of the first input qubit.

$$C_{01} \left|x, y\right\rangle = \left|x \oplus y, y\right\rangle \tag{2.37}$$

$$C_{01} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{2.38}$$

In Figure 2.3 it is represented the typical representation of the gate Controlled-NOT, i.e.,version $C_{10}$ (on the left) and version $C_{01}$ on the right.

Figure 2.3: Representation of the Controlled-NOT gate version $C_{10}$ on the left and version $C_{01}$ on the right

### 2.4.4 Controlled-U Gate

The gate controlled $U$ is described as any unitary operation represented by an unitary matrix $2 \times 2$.

$$U = \begin{pmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{pmatrix} \tag{2.39}$$

This matrix has to be unitary because in quantum mechanics any evolution from a $|\psi\rangle$ to a $|\phi\rangle$ has to preserve the normalization of the quibt, i.e., if we apply one gate to one qubit of norm $1$, the final result has to be a new qubit also with norm $1$.

In (2.40) we have the matrix form of the controlled $U$ gate, which is something very similar to the

controlled NOT but, in this case, we have an arbitrary operation corresponding to matrix $U$.

$$C^U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{pmatrix} \tag{2.40}$$

The $U$ operation is only performed with respect to the second bit if the first bit is $1$. This operation can be, for example, any of the Pauli Gates, resulting in controlled-$X$, controlled-$Y$ or controlled-$Z$. Figure 2.4 shows the representation of Controlled-$U$ gate.

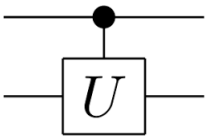



Figure 2.4: Typical representation of Controlled-$U$ gate in quantum computation

### 2.4.5 SWAP Gate

Another important quantum gate is the *SWAP* gate. As the name suggests, this gate allows to swap the order of two qubits.

$$S\left|xy\right\rangle = \left|yx\right\rangle \tag{2.41}$$

If the first input qubit is $x$ and the second input qubit is $y$, the output qubits will be $y$ and $x$, respectively. This is possible by using the matrix represented in (2.42).

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.42}$$

Figure 2.5 represents the SWAP gate in quantum computation.

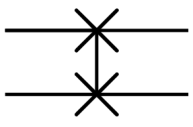


Figure 2.5: Typical representation of SWAP gate in quantum computation

### 2.4.6 Phase Shift Gate

A phase shift gate changes the phase of the qubit. This phase changing performs a rotation along the Bloch sphere. This rotation leaves the latitude intact, by only changing the longitude. The phase

shift gate is represented by the following matrix:

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \tag{2.43}$$

Depending on the value of $\phi$, some predefined gates are determined. For example if $\phi = \frac{\pi}{2}$ we have a $S$ gate and if if $\phi = \frac{\pi}{4}$ we are in the presence of a $T$ gate. The representation of these gates are depicted in Figure (2.6).


Figure 2.6: Representations of a $S$ phase gate and a $T$ phase gate

### 2.4.7 Quantum Parallelism with Quantum Gates

The power of a quantum computer resides in the quantum parallelism associated with quantum superposition. Quantum parallelism is what makes possible to perform a large number of operations in parallel, i.e., simultaneously, which is the huge difference when compared to a classical computer [18].

In Figure 2.7 it is illustrated a black box which is commonly called as oracle in computer science.

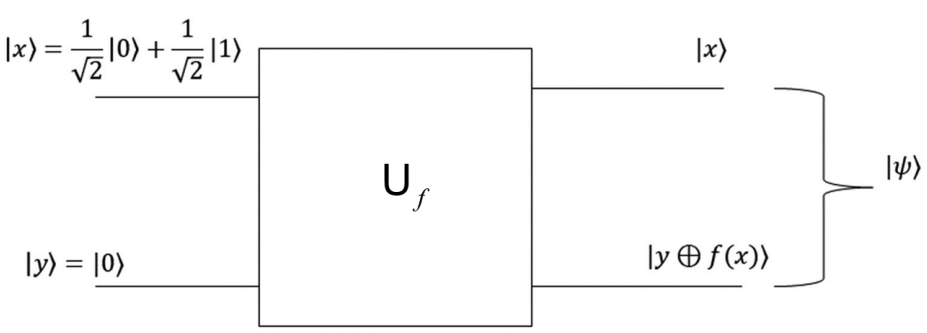



Figure 2.7: Representation of an oracle in a quantum circuit

This oracle is represented as an unitary gate $U_f$ where $f$ is a function. In this example, the first input qubit is $|x\rangle = |\Phi^+\rangle$ and the second qubit $|y\rangle = |0\rangle$. This gate has two output qubits, the first one is equal to the first input $|\Phi^+\rangle$ and the second one corresponds to the modulo-2 addition of the second input qubit and the function $f$ applied to $x$. Of course, in this example, independently of the value of $f(x)$, the result of this addition will always be $f(x)$ since $y = 0$. Considering that the first input is a superposition, it will collapse either for $|0\rangle$ or $|1\rangle$. In other words, the output of the second qubit will be a superposition of $f(0)$ and $f(1)$.

$$\frac{|f(0)\rangle + |f(1)\rangle}{\sqrt{2}} \tag{2.44}$$

This result shows the true power of quantum computing because with the application of only one quantum gate, two arithmetic operations were made simultaneously, $f(0)$ and $f(1)$ and this is the key of

quantum parallelism.

$$\text{Input:} \qquad \frac{|00\rangle + |10\rangle}{\sqrt{2}} \tag{2.45}$$

$$\text{Output:} \qquad |\psi\rangle = \frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}} \tag{2.46}$$

Note that the output state $|\psi\rangle$ contains information about both $f(0)$ and $f(1)$.

In Figure 2.8 we have an application of quantum parallelism.

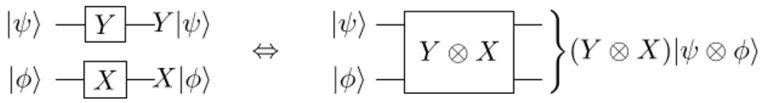



Figure 2.8: Two gates Y and X in parallel is equivalent to the gate $Y \otimes X$

First we have two input qubits, $|\psi\rangle$ and $|\phi\rangle$. Then a gate $Y$ is applied to the first qubit and a gate $X$ is applied to the second qubit. Mathematically speaking, this is the same thing as applying a tensor product of $Y$ with $X$ to the input qubits, which will result in $(Y \otimes X)|\psi \otimes \phi\rangle$.

Another idea of quantum parallelism is the Walsh-Hadamard Transform. This transform can be represented as a $4 \times 4$ matrix when using two qubits.

$$\mathrm{H}^{\otimes 2} = \mathrm{H} \otimes \mathrm{H} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{2.47}$$

$$\mathrm{H}^{\otimes 2}|00\rangle = (\mathrm{H}|0\rangle) \otimes (\mathrm{H}|0\rangle) = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2} \tag{2.48}$$

This matrix can be described as the tensor product of two qubits (applying a Hadamard gate to each one), as described in (2.48), producing an equal superposition of all computational basis states. Again, this is the idea of quantum parallelism: given two qubits, using two Hadamard gates, it transforms a superposition of equal weight into all basis states as shown in Figure 2.9.

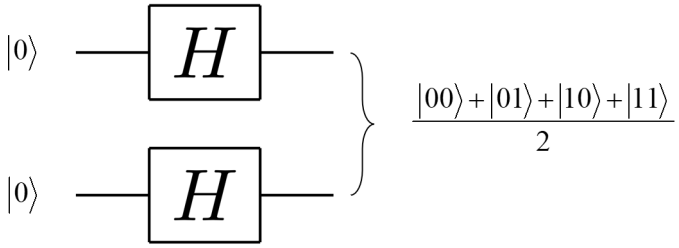



Figure 2.9: Representation of the Walsh-Hadamard transform on two qubits

If instead of two qubits we had $n$ qubits, the tensor product expression would be

$$H^{\otimes n}|\underbrace{00\cdots0}_{n}\rangle = \frac{1}{\sqrt{2^n}}\sum_x |x\rangle. \tag{2.49}$$

## 2.5 Quantum Non-locality with PR Boxes

In 1935 Albert Einstein (1879-1955), Boris Podolsky (1896-1966) and Nathan Rosen (1909-1995) published a paper to study the influence of quantum entanglement in particles and concluded that quantum mechanics was incomplete since they could not prove nor believe in quantum non-locality [21]. Instead, they believed in the existence of *hidden variables*. If two particles were greatly separated from each other, it was impossible to one of them change its state and influence the other particle's state instantaneously. For EPR, the explanation was that the state of these particles were predetermined since their creation, moving aside the idea of non-locality.

Nevertheless, in 1964 an Irish physicist names John Bell proved that it was impossible to exist such hidden variables by using the notorious Bell's Game [22]. This game was then replicated in the 1990's by Sandu Popescu (1956–) and Daniel Rohrlich (1954–) where they used the concept of PR boxes to understand the implications of quantum entanglement [23]. The following section explains the non-local phenomena in quantum mechanics with the use of the notorious PR boxes.

In this game we have two competitors: Alice and Bob. When Alice puts her input $x$ in her PR box, the output will be $a$. On the other hand, when Bob puts his input $y$ in his box, the result of the output will be $b$. Both the inputs and outputs can only assume the values $0$ or $1$. The game has two rules:

1. Every time Alice chooses $x = 0$ or Bob chooses $y = 0$, they score one point if their outputs match (i.e., $a = b$);

2. Every time both Alice and Bob choose $1$ as input (i.e., $x = y = 1$), they score one point if the outputs are different (i.e., $a \neq b$).

With some algebraic manipulation, the rules of this game can be written according to Table 2.1,

| Scoring Conditions | | Alice's input | |
|---|---|---|---|
| | | $a$ = 0 | $a$ = 1 |
| Bob's input | $b$ = 0 | $a \oplus b = x \wedge y$ | $a \oplus b = x \wedge y$ |
| | $b$ = 1 | $a \oplus b = x \wedge y$ | $a \oplus b = x \wedge y$ |

Table 2.1: Scoring conditions

where $\oplus$ is the modulo-2 addition and $\wedge$ is the AND operator. Note that both Alice and Bob can just score one point if the equality in Table 2.1 is verified, which is congruent with the rules of the game.

At this time, it is possible to define two probabilities according to the rules of the game: $P[(a = b) \mid (x, y)]$ for the first rule and $P[(a \neq b) \mid (x, y)]$ for the second rule. According to these probabilities and

Table 2.1, the score of the game $SCORE = S$ is given by

$$S = P[(a = b) \mid (0,0)] + P[(a = b) \mid (0,1)] + P[(a = b) \mid (1,0)] + P[(a \neq b) \mid (1,1)]. \tag{2.50}$$

As stated before, EPR believed that the answers given by Alice and Bob were based on hidden variables, which means that they both have predetermined their answers before the start of the game. While in Table 2.2 it is presented all possible answers that could be given by both players, in Figure 2.10 it is presented a table with all possible $16$ combinations with the local strategies agreed between Alice and Bob.

| Outcomes for Alice | Outcomes for Bob |
|---|---|
| $P1$: Output $a$ = 0 for any input $x$ | $Q1$: Output $b$ = 0 for any input $y$ |
| $P2$: Output $a$ = 1 for any input $x$ | $Q2$: Output $b$ = 1 for any input $y$ |
| $P3$: Output $a$ and input $x$ are equal | $Q3$: Output $b$ and input $y$ are equal |
| $P4$: Output $a$ and input $x$ are opposites | $Q4$: Output $b$ and input $y$ are opposites |

Table 2.2: All predetermined outcomes for Alice and Bob

| | | $[(x, y) = (0, 0)] \Rightarrow (a \oplus b = 0)$ | $[(x, y) = (0, 1)] \Rightarrow (a \oplus b = 0)$ | $[(x, y) = (1, 0)] \Rightarrow (a \oplus b = 0)$ | $[(x, y) = (1, 1)] \Rightarrow (a \oplus b = 1)$ | SCORE |
|---|---|---|---|---|---|---|
| P1 | Q1 | (a, b) = (0, 0) ✓ | (a, b) = (0, 0) ✓ | (a, b) = (0, 0) ✓ | (a, b) = (0, 0) ✗ | 3 |
| P1 | Q2 | (a, b) = (0, 1) ✗ | (a, b) = (0, 1) ✗ | (a, b) = (0, 1) ✗ | (a, b) = (0, 1) ✓ | 1 |
| P1 | Q3 | (a, b) = (0, 0) ✓ | (a, b) = (0, 1) ✗ | (a, b) = (0, 0) ✓ | (a, b) = (0, 1) ✓ | 3 |
| P1 | Q4 | (a, b) = (0, 1) ✗ | (a, b) = (0, 0) ✓ | (a, b) = (0, 1) ✗ | (a, b) = (0, 0) ✗ | 1 |
| P2 | Q1 | (a, b) = (1, 0) ✗ | (a, b) = (1, 0) ✗ | (a, b) = (1, 0) ✗ | (a, b) = (1, 0) ✓ | 1 |
| P2 | Q2 | (a, b) = (1, 1) ✓ | (a, b) = (1, 1) ✓ | (a, b) = (1, 1) ✓ | (a, b) = (1, 1) ✗ | 3 |
| P2 | Q3 | (a, b) = (1, 0) ✗ | (a, b) = (1, 1) ✓ | (a, b) = (1, 0) ✗ | (a, b) = (1, 1) ✗ | 1 |
| P2 | Q4 | (a, b) = (1, 1) ✓ | (a, b) = (1, 0) ✗ | (a, b) = (1, 1) ✓ | (a, b) = (1, 0) ✓ | 3 |
| P3 | Q1 | (a, b) = (0, 0) ✓ | (a, b) = (0, 0) ✓ | (a, b) = (1, 0) ✗ | (a, b) = (1, 0) ✓ | 3 |
| P3 | Q2 | (a, b) = (0, 1) ✗ | (a, b) = (0, 1) ✗ | (a, b) = (1, 1) ✓ | (a, b) = (1, 1) ✗ | 1 |
| P3 | Q3 | (a, b) = (0, 0) ✓ | (a, b) = (0, 1) ✗ | (a, b) = (1, 0) ✗ | (a, b) = (1, 1) ✗ | 1 |
| P3 | Q4 | (a, b) = (0, 1) ✗ | (a, b) = (0, 0) ✓ | (a, b) = (1, 1) ✓ | (a, b) = (1, 0) ✓ | 3 |
| P4 | Q1 | (a, b) = (1, 0) ✗ | (a, b) = (1, 0) ✗ | (a, b) = (0, 0) ✓ | (a, b) = (0, 0) ✗ | 1 |
| P4 | Q2 | (a, b) = (1, 1) ✓ | (a, b) = (1, 1) ✓ | (a, b) = (0, 1) ✗ | (a, b) = (0, 1) ✓ | 3 |
| P4 | Q3 | (a, b) = (1, 0) ✗ | (a, b) = (1, 1) ✓ | (a, b) = (0, 0) ✓ | (a, b) = (0, 1) ✓ | 3 |
| P4 | Q4 | (a, b) = (1, 1) ✓ | (a, b) = (1, 0) ✗ | (a, b) = (0, 1) ✗ | (a, b) = (0, 0) ✗ | 1 |

Figure 2.10: All possible combinations using all different outcomes

Let us take for example the case when Alice uses strategy $P2$ and Bob uses strategy $Q3$. For the first combination of inputs, they are going to disagree because for $a = 1$ and $b = 0$, $a \oplus b = 1$ and not $0$ as stated in beginning of the second column. If we apply the same thought to the rest of the row, we verify that they just score one point out of four with this combination strategy. However, for strategy $P1$ and $Q1$ Alice and Bob get three out of four points and if we the check the SCORE column, we can easily conclude that they just can win at most three out of four points, depending on the combination of strategies used. In other words, it is impossible to win more than three points in the Bell's Game using local strategies. The probability of success to win the game is at most

$$\text{Probability of Success } = \frac{SCORE}{m} \leq \frac{3}{4} = 75\%, \tag{2.51}$$

where $m$ denotes all possible 4 input configurations: (0,0); (0,1); (1,0); (1,1).

Another way to present this game is by using the version Clauser–Horne–Shimony-Holt (CHSH)[5] of Bell's Theorem [24], which is a more adequate way to present results and for that, let us slightly change the previous alphabet. The outputs will now be $A$ and $B$ and can have either $-1$ or $1$ as its values. The terms $A_0$, $A_1$, $B_0$, $B_1$ now refer to Alice's and Bob's inputs (e.g.: $A_0$ means that Alice chose input $x = 0$). The rules of the game are the same and with that, we can clearly see that if the inputs are the same, their product will be positive and if they are different their product will be negative. With this new alphabet, we get two new probabilities: the probability agreement $p$ and disagreement $q$.

$$p = P[(A = B) \mid (x, y)] = P(-1, -1) + P(1, 1) \tag{2.52}$$

$$q = P[(A \neq B) \mid (x, y)] = P(-1, 1) + P(1, -1) \tag{2.53}$$

For every input combination, let us calculate the expected value of the product $AB$ since it corresponds to a correlation between the outputs.

$$\langle AB \rangle = P(-1, -1) - P(-1, 1) - P(1, -1) + P(1, 1) = p - q = 2p - 1 = 1 - 2q \tag{2.54}$$

Note that each term in (2.54) is multiplied by the respective product of outputs between Alice and Bob.

Now we can write the probability of agreement and disagreement in respect to the correlation $AB$.

$$\langle AB \rangle = 2p - 1 \Leftrightarrow p = \frac{1 + \langle AB \rangle}{2} \Rightarrow p_{xy} = \frac{1 + \langle A_x B_y \rangle}{2} \tag{2.55}$$

$$\langle AB \rangle = 1 - 2q \Leftrightarrow q = \frac{1 - \langle AB \rangle}{2} \Rightarrow q_{xy} = \frac{1 - \langle A_x B_y \rangle}{2} \tag{2.56}$$

According to the rules of the game, the score is then given by the sum of three probabilities of agreement and one probability of disagreement.

$$S = p_{00} + p_{01} + p_{10} + p_{11} = 2 + \frac{1}{2} \left[ \langle A_0 B_0 \rangle + \langle A_0 B_1 \rangle + \langle A_1 B_0 \rangle - \langle A_1 B_1 \rangle \right] = 2 + \frac{B}{2} \tag{2.57}$$

Now, according to EPR, Alice's and Bob's inputs and outputs are independent events. This way, we can affirm that the expected value of the product is the product of the expected values. As we have seen in expression (2.57), $B$ is given by

$$\begin{aligned} B = B_{EPR} &= \langle A_0 B_0 \rangle + \langle A_0 B_1 \rangle + \langle A_1 B_0 \rangle - \langle A_1 B_1 \rangle \\ &= \left[ \langle B_0 \rangle + \langle B_1 \rangle \right] \langle A_0 \rangle + \left[ \langle B_0 \rangle - \langle B_1 \rangle \right] \langle A_1 \rangle . \end{aligned} \tag{2.58}$$

If we make $\beta_+ = \langle B_0 \rangle + \langle B_1 \rangle$ and $\beta_- = \langle B_0 \rangle - \langle B_1 \rangle$, equation (2.58) can be written has

$$B_{\mathrm{EPR}} = \beta_+ \langle A_0 \rangle + \beta_- \langle A_1 \rangle . \tag{2.59}$$

[5] In physics, the CHSH inequality is used in the proof of Bell's theorem, which proves that certain consequences of entanglement in quantum mechanics cannot be reproduced by local hidden variables

To find the value of the score, we need to find the value of $B_{EPR}$, and to find the value of $B_{EPR}$ we need to know the values of $\beta_+$ and $\beta_-$. Since $B_0$ and $B_1$ can only assume the values $-1$ or $1$, we can easily see that both $\beta_+$ and $\beta_-$ can only assume the values $-2$, $0$ and $2$.

Given that $A_0$ and $A_1$ can either be $-1$ or $1$, the expected value $B_{EPR}$ must obey

$$-2 \leq B_{EPR} \leq 2, \tag{2.60}$$

and given the expression (2.57)

$$1 \leq S = 2 + \frac{B_{\text{EPR}}}{2} \leq 3. \tag{2.61}$$

Dividing the score by $m$, the probability of success according to EPR is

$$\text{Probability of Success } = \frac{SCORE}{m} = \frac{2 + \frac{B_{\text{EPR}}}{2}}{4} \leq \frac{3}{4} = 75\%, \tag{2.62}$$

which remains the same as before.

Nevertheless, according to quantum mechanics, we can get a higher probability of success without the existence of hidden variables. To prove it, we first need to introduce the Malus' Law which states that the probability of two photons, one from Alice and another from Bob, presenting the same result for any input is given by

$$P[(A = B) \mid (x, y)] = \cos^2(\theta), \tag{2.63}$$

where $\theta$ is the angle between the axis $\theta_1$ of Alice's polarizer and the axis $\theta_2$ of Bob's polarizer[6]. This probability, is of course, the probability of agreement and, therefore, the probability of disagreement according to Pythagoras is given by

$$P[(A \neq B) \mid (x, y)] = 1 - \cos^2(\theta) = \sin^2(\theta). \tag{2.64}$$

Note that when the angle between the axes is $0$, there is $100\%$ of probability of agreement, which means the photons pass through the polarizer. If the axes are orthogonal, i.e., 90º, this probability becomes $0\%$, which means that no photon passes through the polarizer (similarly with the experiment presented in Chapter 1).

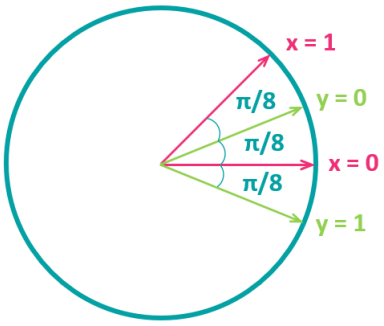



Figure 2.11: Different angles of polarization according to the inputs

[6] $\theta = \theta_1 - \theta_2$

Let us now focus our attention to Figure 2.11. When Alice chooses input $x = 0$ and changes her angle of polarization to $x = 1$, it makes an angle of $2 \times \frac{\pi}{8} = \frac{\pi}{4}$ with the first direction. The same applies to Bob. With that being said, we can calculate the angle $\theta$ for the four different configurations:

$$1^{\text{st}} \text{ Configuration: } (x = 0, y = 0) \Rightarrow \theta = \frac{\pi}{8} \tag{2.65}$$

$$2^{\text{nd}} \text{ Configuration: } (x = 0, y = 1) \Rightarrow \theta = \frac{\pi}{8} \tag{2.66}$$

$$3^{\text{rd}} \text{ Configuration: } (x = 1, y = 0) \Rightarrow \theta = \frac{\pi}{8} \tag{2.67}$$

$$4^{\text{th}} \text{ Configuration: } (x = 1, y = 1) \Rightarrow \theta = \frac{3\pi}{8} \tag{2.68}$$

To calculate the score according to quantum mechanics, we need to also calculate the mean value $B$. For that, let us determine the non-linear correlation between the product of $A$ and $B$ for all possible cases.

$$\begin{aligned} \langle A_0 B_0 \rangle &= P[(A = \mathrm{B}) \mid (0,0)] - P[(A \neq \mathrm{B}) \mid (0,0)] \\ &= \cos^2(\theta) - \sin^2(\theta) = \cos^2\left(\frac{\pi}{8}\right) - \sin^2\left(\frac{\pi}{8}\right) = \frac{1}{\sqrt{2}} \\ &= \langle A_0 B_1 \rangle = \langle A_1 B_0 \rangle \end{aligned} \tag{2.69}$$

$$\begin{aligned} \langle A_1 B_1 \rangle &= P[(A = B) \mid (1,1)] - P[(A \neq B) \mid (1,1)] \\ &= \cos^2(\theta) - \sin^2(\theta) = \cos^2\left(\frac{3\pi}{8}\right) - \sin^2\left(\frac{3\pi}{8}\right) = -\frac{1}{\sqrt{2}} \end{aligned} \tag{2.70}$$

Since the addition of these four results brings

$$B_{QM} = \langle A_0 B_0 \rangle + \langle A_0 B_1 \rangle + \langle A_1 B_0 \rangle - \langle A_1 B_1 \rangle = 2\sqrt{2} \tag{2.71}$$

and the score given by (2.57) can be written as

$$SCORE = 2 + \frac{1}{2}\,\mathrm{B_{QM}} = 2 + \frac{1}{2}(2\sqrt{2}) = 2 + \sqrt{2} \approx 3.4142, \tag{2.72}$$

the probability of success in quantum mechanics is then given by

$$\text{Probability of Success } = \frac{SCORE}{m} = \frac{2\sqrt{2}}{4} \approx 85.36\%. \tag{2.73}$$

This result proves that, without hidden variables, it is possible to obtain a higher probability of success of winning the Bell's game when comparing to EPR. Local realism defended by EPR is not observed meaning that it does not exist any hidden variables. Instead, we can affirm the existence of quantum non-locality.

## 2.6 The Hardy State

Another interesting example to analyze the divergence of results between EPR and quantum mechanics is an experiment proposed by the physicist Lucien Hardy (1966-). We start by introducing expression (2.74),

$$|\Phi\rangle = \frac{1}{\sqrt{12}}(3|00\rangle + |01\rangle + |10\rangle - |11\rangle) \tag{2.74}$$

where $|\Phi\rangle$ is a Hardy state[25, Appendix B 6.6]. Note that this state is already normalized, meaning that the sum of the weights of the squares from each *ket* is equal to $1$. Suppose now that both Alice and Bob have one qubit of the entangled pair from (2.74), which can also be written[7] as

$$|\Phi\rangle = \frac{1}{\sqrt{3}}\left(2|00\rangle - H_A H_B|11\rangle\right), \tag{2.75}$$

where $H_A$ represents the scenario when Alice applies a Hadamard gate to her qubit and $H_B$ when Bob applies a Hadamard gate to his. To decide when they should apply a Hadamard gate (or not) before measuring the qubit, Alice and Bob toss a coin to the air. At this point, four different events can happen:

(i) Neither Alice nor Bob apply a Hadamard gate, therefore, according to (2.74), they both get the result $1$ with a probability of $\left(\frac{1}{\sqrt{12}}\right)^2 = \frac{1}{12}$;

(ii) Only Alice applies a Hadamard gate to her qubit, resulting in the state[8]

$$\begin{aligned} H_A|\Phi\rangle &= \frac{1}{\sqrt{3}}\left(2H_A|00\rangle - H_A H_A H_B|11\rangle\right) = \frac{1}{\sqrt{3}}\left(2H_A|00\rangle - H_B|11\rangle\right) \\ &= \frac{1}{\sqrt{24}}(4|00\rangle + 2|10\rangle + 2|11\rangle); \end{aligned} \tag{2.76}$$

(iii) Only Bob applies a Hadamard gate to his qubit, resulting in the state

$$\begin{aligned} H_B|\Phi\rangle &= \frac{1}{\sqrt{3}}\left(2H_B|00\rangle - H_A H_B H_B|11\rangle\right) = \frac{1}{\sqrt{3}}\left(2H_B|00\rangle - H_A|11\rangle\right) \\ &= \frac{1}{\sqrt{24}}(4|00\rangle + 2|01\rangle + 2|11\rangle); \end{aligned} \tag{2.77}$$

(iv) Both Alice and Bob apply a Hadamard gate[9] to their qubits. In this case, the state of the qubits will be

$$\begin{aligned} H_A H_B|\Phi\rangle &= \frac{1}{\sqrt{3}}\left(2H_A H_B|00\rangle - H_A H_A H_B H_B|11\rangle\right) \\ &= \frac{1}{\sqrt{3}}\left(2\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) - |11\rangle\right) \\ &= \frac{1}{\sqrt{3}}\left(2 \times \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) - |11\rangle\right) \\ &= \frac{1}{\sqrt{3}}(|00\rangle + |01\rangle + |10\rangle). \end{aligned} \tag{2.78}$$

According to these four events (i.e., according to quantum mechanics), it is possible to build Table 2.3.

[7] This equality is demonstrated in Appendix B.1

[8] Remember equation (2.32) of Chapter 2, where $H^2 = 1$

[9] An example of application of a Hadamard gate for a two qubit state is demonstrated in Appendix B.2

| Gates | | Results | | Possible? |
|---|---|---|---|---|
| Alice | Bob | Alice | Bob | Yes/No |
| I | I | 1 | 1 | Yes |
| H | I | 0 | 1 | No |
| I | H | 1 | 0 | No |
| H | H | 1 | 1 | No |

Table 2.3: Likelihood of events according to quantum mechanics

On the first row of the table (case (i)), Alice and Bob do nothing to their qubits. We have seen that, according to quantum mechanics, this event is possible (with a probability of $\frac{1}{12}$) since state $|\Phi\rangle$ from (2.74) stays intact and the result $1$ for Alice and $1$ for Bob ($|11\rangle$) is contained in the equation. In case (ii), when Alice applies a Hadamard gate to her qubit, we get expression (2.76), which clearly states that the result $0$ for Alice and $1$ for Bob ($|01\rangle$) after measurement is not possible. On third row Bob applies a Hadarmard gate to his qubit, resulting in (2.77). Similarly of what happened in the second case, the result of this third case, $1$ for Alice and $0$ for Bob ($|10\rangle$) is not possible. On the fourth and final case they both apply a Hadamard gate to their qubit, and the probability of both getting a $1$ as result after measurement is $0$ since $|11\rangle$ does not appear in (2.78).

Nevertheless, one could argue according to EPR to interpret this experiment. Would they come to the same results? Note that for EPR there is not "spooky action at the distance", meaning that the result of Alice measurement cannot influence Bob's result and vice-versa and, if it does, it would have been with the presence of hidden variables.

In the first case they both toss a coin and nobody applies the Hadamard gate, for which the result is $1$ for both. This probability is therefore $\frac{1}{4} \times \frac{1}{12} = \frac{1}{48}$. In the second case Alice applies a Hadamard and the result $0$ for Alice and $1$ for Bob is not possible. Therefore, the only possible result is $1$ for Alice and $1$ for Bob (since Bob did nothing to his qubit). In case (iii) it is applied the same reasoning. The result $1$ for Alice and $0$ for Bob is not possible and since Alice did not do anything to her qubit, the only possible result is $1$ for Alice and $1$ for Bob. So far so good, it seems that both EPR and quantum mechanics "agree" in the results.

Now, according to EPR, we can state the following: independently of the result of the measurement for the (iv) case, where both players apply a Hadamard gate, based on the previous three cases, it is "clear" that the only possible solution for that outcome is $1$ for Alice and $1$ for Bob. But this statement contradicts quantum mechanics's conclusion. We have here yet another contradiction between these two reasonings.

In summary quantum mechanics states that in the first case where nobody applies a Hadamard gate is possible to get the result $1$ for Alice and $1$ for Bob. According to EPR nothing prevents this result so it is possible too. However, for the second and third case where just one Hadamard gate is applied, quantum mechanics states that the result $1$ for Alice and $0$ for Bob and the result $0$ for Alice and $1$ for Bob are impossible. Note that it is not presented a solution or a probable result of what should happen;

it is just said, with all certain, that those results are guaranteed not to happen. On the other hand, EPR defends that if those results are impossible, the only coherent result must be $1$ for both Alice and Bob (if a Hadamard gate is applied to just one qubit in the state $1$ and the result cannot be $0$, it can only be $1$). Based on these previous conclusions and guided by the motto that it does not exist ”spooky action at the distance”, the fourth case can only be $1$ for Alice and $1$ for Bob, which quantum mechanics states and mathematically proves that that outcome cannot happen. For quantum mechanics each case must be treated independently whereby assuming a result based on previous results is not the correct procedure to think when dealing with quantum measurements. As the physicist David Mermin (1935-) once said: ”What did not happen, did not happen”.

# Chapter 3

# Quantum Circuits and Quantum Algorithms

This chapter is centered on the essential bases that rule quantum computation. The foundations of quantum computing, introduced in the previous chapter, will now be put into practice throughout quantum circuits and algorithms, namely, the Simon's algorithm. It starts to cover a classical algorithm named RSA before jump into quantum algorithms and their properties, so it is perceptible the differences between classical and quantum approaches.

## 3.1 RSA Algorithm

RSA algorithm[26, Chapter 3] [15, Appendix 5][25, Chapter 3] is an asymmetric cryptography algorithm that is widely used for secure data transmission. The acronym "RSA" are the initials of the three scientists who invented this cryptographic system back in 1977: Ron Rivest (1947-), Adi Shamir (1952-) and Leonard Adleman (1945-).

The RSA is called an asymmetric algorithm because it works with the use of two different keys, a public key and a private key, making the change of messages over the internet between two or more persons very secure. While the public key serves for encryption and is given to the public, the private key is used for decryption and is kept only by the person that sends a message. This way, only the intended receiver can decrypt and read the message.

The security and the difficulty to break this algorithm relies on the generation of a natural number with hundreds of digits and then factorizing that number into the product of two large and different prime numbers. It would be very difficult for an eavesdropper to figure out which prime numbers must be multiplied to discover the big non-prime number and decipher the message because it would take a very long time to reverse engineer the method. Even if the eavesdropper uses a supercomputer to do many guesses per second, the process would still be very slow, possibly taking years if not centuries to discover the correct solution.

### 3.1.1 Theory Behind the RSA Algorithm

Alice and Bob want to communicate with each other using encrypted messages. Bob will firstly generate two different prime numbers, $p$ and $q$, and calculate the product of those numbers, which will result in a very big integer composite number $N$.

$$N = pq \tag{3.1}$$

Then, he will select a number $e$ which he will use for encryption and a number $d$ which he will use for decryption. Note that $d$ is a private key, therefore only Bob knows information about it. Not even Alice can decrypt it because she does not have Bob's private key.

The number $e$ (which is public) is chosen in such a way that it must satisfy (3.2)

$$1 < e < \phi(N), \tag{3.2}$$

where $\phi(N)$ is the Euler function[1]. Since $N$ is the product of two prime numbers, $\phi(N)$ will be given by (3.3)

$$\phi(N) = (p-1)(q-1). \tag{3.3}$$

Besides that, $e$ and $\phi(N)$ must be co-primes, i.e.,

$$\text{gcd}(e, \phi(N)) = 1. \tag{3.4}$$

The number $d$ (which is private) is chosen in such a way that it must satisfy (3.5) and (3.6).

$$1 < d < \phi(N) \tag{3.5}$$

$$ed \equiv 1 (\text{mod } \phi(N)) \tag{3.6}$$

On the other hand, Alice wants to send Bob a message $M$, but she will not transmit $M$ trough the public channel. Instead, she will first use an encryption process to code her message making it an encrypted text, which will result in $C$.

$$C = M^e (\text{mod } N) \tag{3.7}$$

Bob will receive the encrypted message $C$ and will decrypt it with the secret key $d$ which only him has. After that Bob (and only Bob) can read the original message given by Alice.

Note that even if an eavesdropper makes their own copy of the public key, it will be useless because the attacker cannot decrypt messages that were sent by Alice. That would require Bob's private key, something that only Bob has. However, given enough time, an eavesdropper's computer can create an exact copy of Bob's private key by guessing many potential private keys, i.e., the correct two prime numbers, until a match is found. Fortunately, this scenario is very unlikely because it could take decades

[1] In number theory, Euler's totient function counts the positive integers up to a given integer $n$ that are relatively prime to $n$.

to find those numbers. The eavesdropper would not find Bob's private key unless Bob was astronomically unlucky.

### 3.1.2 RSA Algorithm: Example

In order to better understand the RSA algorithm, let us see the following example. For sake of simplicity, this example will work with very small numbers which is a very unrealistic example.

Let us suppose that Bob chooses the prime numbers $p = 11$ and $q = 23$. The product $N$ will then be given by

$$N = 253. \tag{3.8}$$

The Euler function correspondent to $N = 253$ is

$$\phi(N) = (11 - 1)(23 - 1) = 10 \times 22 = 220. \tag{3.9}$$

Now Bob must choose a number $e$ which must obey (3.2), that is,

$$1 < e < 220. \tag{3.10}$$

Let us suppose he chose 81. Note that this number cannot be absolutely arbitrary because the greatest common divisor of 81 and 220 must be 1 as stated in (3.4). In other words, in number theory, 81 and 220 must be co-primes.

Once $e$ is chosen, Bob must choose his secret key $d$. Substituting $e$ and $\phi(N)$ in (3.6), we obtain

$$81d \equiv 1(\text{mod } 220) \Leftrightarrow d = 81^{-1}(\text{mod}220) = 201, \tag{3.11}$$

which is the secret key that only Bob knows about.

Alice now will receive the public key $(N, e) = (253, 81)$ and transmit her secret message $M$ to Bob. Note that this message must obey (3.12)

$$0 < M < N = 253. \tag{3.12}$$

Let us suppose that $M = 200$. She then encrypts $M$, resulting in $C$

$$C = 200^{81}(\text{mod } 253) = 101. \tag{3.13}$$

This number $C$ will be sent publicly to Bob. Bob then receives it and uses $d$ to decrypt the message

$$C^d(\text{mod } N) = 101^{201}(\text{mod } 253) = 200. \tag{3.14}$$

The decrypted message is exactly the message sent by Alice, $M = 200$.

This is how RSA algorithm works. Although it is a very reliable and secure way to encrypt and

decrypt data since a classical computer would take exponential time to discover the message, a quantum computer could easily break RSA algorithm in polynomial time, which is way faster.

## 3.2 No-Cloning Theorem

The following section describes an important consequence of unitary condition: it is impossible to clone a quantum state [17, Section 5.1.1].

One should note that this "cloning" procedure is not the same as cloning in biology where two organisms share the same DNA. Cloning in physics means a more "perfect" copy: where the relative positions and momenta and energy levels of every particle are exactly the same in both the original and the copied one.

Before we dive any deeper into this topic, it is relevant to point out three important properties that all quantum particles can have. The first one is superposition, where the whole is equal to the sum of its different possible parts.

$$|C\rangle = |C_1\rangle + |C_2\rangle \tag{3.15}$$

The second property happens when groups or combinations of particles are described as products of their components (or sums of products of their components if it is in a superposition). Let us take the example of Schrödinger's cat. In 1935 Erwin Schrödinger (1887-1961) proposed a thought experiment[27] where a cat is locked in a box and this cat's life or death state depends on the state of a radioactive atom whether it emitted radiation or not. According to Schrödinger, the Copenhagen interpretation[2] implies that if we do not open the box and measure the state of the cat, is as if it was in a state of both dead and alive. This phenomenon can be described as

$$|\text{State of the Cat}\rangle = |\text{Radiation emitted }\rangle|\text{Cat is dead}\rangle + |\text{Radiation not emitted}\rangle|\text{Cat is alive}\rangle \tag{3.16}$$

The third property states that if a particle is in a superposition and it is changed, this change is replicated in all states independently.

$$T\left(|C_1\rangle + |C_2\rangle\right) = T\left(|C_1\rangle\right) + T\left(|C_2\rangle\right) \tag{3.17}$$

Now that we have these properties in mind, let us see an example. For this example, we will not use quantum states. Instead, we are going to use macro-objects for a more intuitive and clear explanation. Note that the conclusions by either using a macro-example or quantum states are the same.

Alice and Bob want to make an exact copy of their favorite signed music album. Alice chose her album of ABBA, which we are going to denominate as $A$ and Bob chose his album of The Beatles, which we are a going to call $B$. To make a copy they need, of course, the music album ($A$ or $B$), a raw material (designated as $|0\rangle$) and the procedure to clone the album by transforming the raw material in a new but identical album. The question now is: is Alice's and Bob's cloning task possible?

[2]The Copenhagen interpretation is one of the oldest proposed interpretations of quantum mechanics, developed by Niels Bohr (1885-1962) and Werner Heisenberg (1901-1976).

Let us assume it is possible and therefore:

$$\text{Clone}\ (|A\rangle \otimes |0\rangle) = |A\rangle \otimes |A\rangle \tag{3.18}$$

$$\text{Clone}\ (|B\rangle \otimes |0\rangle) = |B\rangle \otimes |B\rangle \tag{3.19}$$

By superposition of both (3.18) and (3.19), we can clone the sum of these situations (third property) with the only difference that we are going to multiply $|A\rangle$ by the complex number $\alpha$ and $|B\rangle$ by the complex number $\beta$ since any process in quantum mechanics should be linear.

$$\begin{aligned} \text{Clone}[(\alpha|A\rangle + (\beta|B\rangle) \otimes |0\rangle] &= \alpha\text{Clone}(|A\rangle \otimes |0\rangle) + \beta\text{Clone}(|B\rangle \otimes |0\rangle) \\ &= \alpha(|A\rangle \otimes |A\rangle) + \beta(|B\rangle \otimes |B\rangle) \end{aligned} \tag{3.20}$$

On the other hand, if we simply apply the cloning process without making use of the third property, we get

$$\begin{aligned} \text{Clone}[(\alpha|A\rangle + (\beta|B\rangle) \otimes |0\rangle] &= (\alpha|A\rangle + \beta|B\rangle) \otimes (\alpha|A\rangle + \beta|B\rangle) \\ &= \alpha^2(|A\rangle \otimes |A\rangle) + \alpha\beta(|A\rangle \otimes |B\rangle) + \beta\alpha(|B\rangle \otimes |A\rangle) + \beta^2(|B\rangle \otimes |B\rangle) \end{aligned} \tag{3.21}$$

Confronting the right-hand side of both equations, it was expected to get the same result since the left-hand side is the same for both. However, that is not the case. The sum of the two terms in (3.20) is not the same as the sum of the four terms in (3.21), unless $\alpha = 0$ and $\beta = 0$. What this example basically states is that if both quantum mechanics and cloning are true, the proposition $(a+b)^2 = a^2 + b^2$ is also true. Nevertheless, that is not correct since $(a+b)^2 = a^2 + 2ab + b^2$

We have found a contradiction and consequently we can conclude that it is impossible to make an exact copy of Alice's and Bob's albums and, therefore, it is impossible to clone an arbitrary quantum state without first destroying the original.

In contrast to cloning, there is no problem transporting arbitrary quantum states from one system to another as we will see further.

## 3.3 Superdense Coding

In this section we will introduce a quantum computation operation that increases the information capacity of a channel by transmitting two classical bits using a single qubit. This operation is called *superdense coding* protocol. To understand this protocol let us presume three different assumptions:

- There is perfect classical communication;
- There is perfect quantum communication
- There is perfect entanglement.

Imagine that Alice and Bob are far away from one another, and Alice wants to send two classical bits to Bob ($00$, $01$, $10$ or $11$) using a single qubit. They are both sharing a pair of qubits in the entangled state

$\Phi^+$ (2.11), where the first qubit is in Alice's possession and the second qubit in Bob's. How can Alice send her information with just one qubit?

Before she actually sends any information, Alice applies some quantum gates in her qubit, depending on the information she wants to send. If she intends to send the bit string '$00$' to Bob, then she does nothing to her qubit.

$$(\mathrm{I}\otimes\mathrm{I})\left|\Phi^+\right\rangle = (\mathrm{I}\otimes\mathrm{I})\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{|00\rangle+|11\rangle}{\sqrt{2}} = \left|\Phi^+\right\rangle \tag{3.22}$$

If she intends to send the bit string '$01$', then she applies the quantum gate $X$ to her qubit. By doing this she will apply a NOT operation to the first qubit meaning that the first qubit '$0$' will turn into '$1$' and '$1$' into '$0$'.

$$(\mathrm{X}\otimes\mathrm{I})\left|\Phi^+\right\rangle = (\mathrm{X}\otimes\mathrm{I})\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{|10\rangle+|01\rangle}{\sqrt{2}} = \left|\Psi^+\right\rangle \tag{3.23}$$

If she wants to send '$10$' she applies the $Z$ gate to her qubit and consequently, she performs a phase flipping, changing the signal $+1$ to $-1$.

$$(\mathrm{Z}\otimes\mathrm{I})\left|\Phi^+\right\rangle = (\mathrm{Z}\otimes\mathrm{I})\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{|00\rangle-|11\rangle}{\sqrt{2}} = \left|\Phi^-\right\rangle \tag{3.24}$$

Finally if she wishes to send the string '$11$' she applies a $-iY$ gate to her qubit, which is the same thing as applying both the $X$ gate and the $Z$ gate.

$$(-i\mathrm{Y}\otimes\mathrm{I})\left|\Phi^+\right\rangle = (-i\mathrm{Y}\otimes\mathrm{I})\frac{|00\rangle+|11\rangle}{\sqrt{2}} = \frac{-|10\rangle+|01\rangle}{\sqrt{2}} = \left|\Psi^-\right\rangle \tag{3.25}$$

After she applied the quantum gate pretended, Alice sends her qubit to Bob through a noiseless quantum channel.

Now that Bob has received her state, he has to decode the information and for that he applies a Controlled-NOT to the two qubits of the entangled pair (changing the second bit of each qubit if the first qubit is 1) and then applies a Hadamard gate to the first qubit.

$$\begin{cases} |\Phi^+\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}} \\ |\Psi^+\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}} \\ |\Phi^-\rangle = \frac{|00\rangle-|11\rangle}{\sqrt{2}} \\ |\Psi^-\rangle = \frac{|01\rangle-|10\rangle}{\sqrt{2}} \end{cases} \xrightarrow{C_{NOT}} \begin{cases} \frac{|00\rangle+|10\rangle}{\sqrt{2}} \\ \frac{|01\rangle+|11\rangle}{\sqrt{2}} \\ \frac{|00\rangle-|10\rangle}{\sqrt{2}} \\ \frac{|01\rangle-|11\rangle}{\sqrt{2}} \end{cases} \Leftrightarrow \begin{cases} \frac{|0\rangle+|1\rangle}{\sqrt{2}}\otimes|0\rangle \\ \frac{|0\rangle+|1\rangle}{\sqrt{2}}\otimes|1\rangle \\ \frac{|0\rangle-|1\rangle}{\sqrt{2}}\otimes|0\rangle \\ \frac{|0\rangle-|1\rangle}{\sqrt{2}}\otimes|1\rangle \end{cases} \Leftrightarrow \begin{cases} |+\rangle\otimes|0\rangle \\ |+\rangle\otimes|1\rangle \\ |-\rangle\otimes|0\rangle \\ |-\rangle\otimes|1\rangle \end{cases} \xrightarrow{\mathrm{H}\otimes\mathrm{I}} \begin{cases} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{cases} \tag{3.26}$$

By doing this, Bob can decode the message getting the states $00$, $01$, $10$ or $11$ depending on Alice's previous choices.

## 3.4 Quantum Teleportation

In this section we are going to discuss *quantum teleportation*[3] [28, Section 9.5]. The objective of quantum teleportation is to transmit the exact same information about a quantum state from a source to a target, by only using classical bits (in contrast with superdense coding) through a communication channel. But according to the no-cloning theorem, we cannot simply copy a qubit's state from one place to another without destroying the original state. However, quantum teleportation does not try to clone a qubit's state. Instead, it reconstructs at the target, with the sent bits, the exact quantum state at the source (which has not been preserved). Let us see an example for a better understanding on this matter.

In this new experiment, Alice and Bob are greatly separated from each other and have the same entangled state that they had in the previous section (2.11), i.e., one entangled qubit is with Alice and another with Bob. However, this time Alice has another qubit in the state $|\psi\rangle$. Alice wants to teleport this qubit to Bob. Worth noting that Bob has no information about this qubit.

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle \tag{3.27}$$

This way, the starting state is the three-qubit quantum state,

$$\begin{aligned} |\psi\rangle \otimes \left|\Phi^+\right\rangle &= \frac{1}{\sqrt{2}}(a|0\rangle + b|1\rangle) \otimes (|00\rangle + |11\rangle) \\ &= \frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|100\rangle + b|111\rangle), \end{aligned} \tag{3.28}$$

where the first two qubits (left to right) are controlled by Alice and the last qubit is controlled by Bob.

Now Alice uses a decoder (similarly to what Bob has done in superdense coding) in her three-qubit state, i.e., she applies a CNOT gate followed by a Hadamard gate[4].

$$\begin{aligned} &(\mathrm{H} \otimes \mathrm{I} \otimes \mathrm{I})\,(\mathrm{U}_{\mathrm{CNOT}} \otimes \mathrm{I})\left(|\psi\rangle \otimes \left|\Phi^+\right\rangle\right) \\ =&(\mathrm{H} \otimes \mathrm{I} \otimes \mathrm{I})\frac{1}{\sqrt{2}}(a|000\rangle + a|011\rangle + b|110\rangle + b|101\rangle) \\ =&\frac{1}{2}[a(|000\rangle + |011\rangle + |100\rangle + |111\rangle) + b(|010\rangle + |001\rangle - |110\rangle - |101\rangle)] \\ =&\frac{1}{2}[|00\rangle(a|0\rangle + b|1\rangle) + |01\rangle(a|1\rangle + b|0\rangle) + |10\rangle(a|0\rangle - b|1\rangle) + |11\rangle(a|1\rangle - b|0\rangle)] \end{aligned} \tag{3.29}$$

At this time, Alice performs a measurement to her two qubits obtaining the four basis states with equal probability. According to (3.29), if she measures $|00\rangle$, Bob's qubit state will be $a|0\rangle + b|1\rangle$. If she measures $|01\rangle$, Bob's qubit state will be $a|1\rangle + b|0\rangle$. If her measurement is $|10\rangle$ Bob's qubit is at state $a|0\rangle - b|1\rangle$. Finally, is Alice measures $|11\rangle$, he gets a qubit in state $a|1\rangle - b|0\rangle$.

Now Alice sends the result of her measurement to Bob using, for example, an optical fiber. Now Bob has all the information needed to reconstruct Alice's qubit $|\psi\rangle$, even if he does not have it. As stated before, depending on Alice measurement, Bob's qubit state will collapse to one of the available options. Nevertheless, Bob still needs to apply a gate to his qubit in order to recover qubit $|\psi\rangle$. If Bob receives

[3] Even if the topic seems sci-fi, what is described is not fiction. Quantum teleportation has already been performed in laboratory.
[4] Remember expression (2.33) which applies a Hadamard gate to three qubits.

$|00\rangle$, he applies an identity gate, which is the same as not applying anything. This way, he obtains indeed $|\psi\rangle$.

$$\mathrm{I}(a|0\rangle + b|1\rangle) = a|0\rangle + b|1\rangle = |\psi\rangle \tag{3.30}$$

If Bob receives $|01\rangle$, he should apply a $X$ gate to obtain $|\psi\rangle$.

$$\mathrm{X}(a|1\rangle + b|0\rangle) = a|0\rangle + b|1\rangle = |\psi\rangle \tag{3.31}$$

When Bob receives $|10\rangle$, he applies a $Z$ gate to his qubit.

$$\mathrm{Z}(a|0\rangle - b|1\rangle) = a|0\rangle + b|1\rangle = |\psi\rangle \tag{3.32}$$

At last, if Bob receives $|11\rangle$, he must apply a $X$ gate and a $Z$ gate (or simply apply $-i\mathrm{Y}$) in order to correct the bit flip and the phase flip that occurred in his qubit so he can recover the state $|\psi\rangle$.

$$-i\mathrm{Y}(a|1\rangle - b|0\rangle) = a|0\rangle + b|1\rangle = |\psi\rangle \tag{3.33}$$

In summary, while in quantum superdense coding it is necessary to have a quantum channel to send one qubit from Alice to Bob so he can receive the information about classical bits, in quantum teleportation, Alice uses a classical communication channel to send her classical bits to Bob, so he can teleport the qubit in her possession.

Quantum teleportation can be very useful since it can transmit a qubit to one point to another. It is also used to reduce computational errors, to form networks of quantum computers and it can improve our security by creating ultra-secure communications channels.

## 3.5 The BB84 Protocol

The security on the internet depends on encryption. It allows us to send private messages to anyone we want without anyone eavesdropping. The encryption algorithms of almost all private information sent over the internet is based on one mathematical process: factorization. The most famous of those algorithms is the Rivest–Shamir–Adleman (RSA) algorithm. A classic computer could take years and a huge amount of computing resources to break RSA encryption. In other words, RSA encryption is not invincible; it is just a really hard problem to solve. Nevertheless, a quantum computer can be used to break some of the best public key cryptosystems, including the RSA algorithm (for example, with the notorious Shor's Algorithm (See Appendix C)). And not just that. Even if it seems a potential threat to our security by using qubits instead of classical bits to take down the best security algorithms we know today, we are completely safe from eavesdroppers due to some quantum proprieties we will see in this section with the explanation of one of the best cryptographic protocols created: the Bennett-Brassard-1984 (BB84) protocol.

The *BB84 protocol* [29][30][28, Section 9.2] was created by Charles Bennett (1943-) and Gilles

Brassard (1955-) in 1984. In quantum cryptography it is possible for two parties to share random "secret keys" to encrypt and decrypt messages between them. This "secret key" cryptographic protocol is usually called quantum key distribution and BB84 was the first quantum key exchange protocol ever created. To better understand the idea of secret keys and how this protocol works let us see an example.

Imagine that Alice wants to send a key message to Bob. For that, she first produces a random sequence of classical bits and then, for each bit, she produces a qubit and polarizes it in one of the following orthogonal bases:

| Basis\State | $\|0\rangle$ | $\|1\rangle$ |
|---|---|---|
| + | $\rightarrow$ | $\uparrow$ |
| $\times$ | $\nearrow$ | $\searrow$ |

Table 3.1: Qubit's polarization direction depending on the bases $+$ and $\times$

Given the basis $+$, a vertical polarization $\rightarrow$ corresponds to $|0\rangle$ and a horizontal polarization $\uparrow$ corresponds to the state $|1\rangle$. On the other hand, given the basis $\times$, a polarization at $45°$ ($\nearrow$) corresponds to the state $|0\rangle$ and a polarization at $135°$ or $-45°$ ($\searrow$) corresponds to the state $|1\rangle$. For example, if Bob measures $\uparrow$ with the $+$ basis and Alice has sent a $\uparrow$, he should get a $|1\rangle$. But what if Alice sends a $\nearrow$ and Bob still uses the $+$ basis to measure Alice's qubit? In that case, Bob will have a superposition of states $\rightarrow$ and $\uparrow$ which will collapse after his measurement to $|0\rangle$ or $|1\rangle$ with a probability of $50\%$ each. In (3.34) is listed all possible superpositions given the bases $+$ and $\times$.

$$\begin{aligned} &\text{If a } + \text{ basis is chosen: } \begin{cases} |\searrow\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle - \frac{1}{\sqrt{2}}|\rightarrow\rangle \\ |\nearrow\rangle = \frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\rightarrow\rangle \end{cases} \\ &\text{If a } \times \text{ basis is chosen: } \begin{cases} |\uparrow\rangle = \frac{1}{\sqrt{2}}|\nearrow\rangle + \frac{1}{\sqrt{2}}|\searrow\rangle \\ |\rightarrow\rangle = \frac{1}{\sqrt{2}}|\nearrow\rangle - \frac{1}{\sqrt{2}}|\searrow\rangle \end{cases} \end{aligned} \tag{3.34}$$

Alice produces a sequence of bits, polarize them according to a certain basis of her choice and send the key message produced to Bob.

| Alice's random bits | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|
| Alice's random bases | $\times$ | + | + | $\times$ | + | $\times$ | $\times$ | + |
| Information sent by Alice | $\searrow$ | $\uparrow$ | $\rightarrow$ | $\nearrow$ | $\rightarrow$ | $\searrow$ | $\nearrow$ | $\uparrow$ |

Table 3.2: Generation of Alice's key message

After that, it is time for Bob to make his measurements. Since he does not know which bases were chosen by Alice, he chooses his own bases for each qubit he receives. When Bob finishes to decode the bit-stream, he will communicate with Alice through a public channel so they can check which basis each one used. Bob will keep the bit he if his base and Alice's base are the same and, if they used different basis, Bob scratches out that bit for further correction. This checking process will result in a smaller sequence of bits that were sent and received in the same basis. If Alice and Bob are safe from any intruders, the sub-sequence of bits of both must match since they have used the same bases and subsequently get the same polarization (which it does as we can see in Table 3.3). On the final row we

have in green the exact bits from Alice due to the use of the same basis and in red the incorrect bits which shows that they used different basis. In other words, the outcomes of both Alice and Bob are identical if the measurement bases are the same, otherwise they are identical with a probability of $50\%$.

| Alice's random bits | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|
| Alice's random bases | × | + | + | × | + | × | × | + |
| Information sent by Alice | ↘ | ↑ | → | ↗ | → | ↘ | ↗ | ↑ |
| Bob's random bases | + | + | × | × | + | × | + | + |
| Bob's random polarizations | → | ↑ | ↘ | ↗ | → | ↘ | ↑ | ↑ |
| Same basis? | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ |
| Shared Secret Key | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

Table 3.3: Checking process so Bob can determine Alice's original secret key

But what if a third party is spying on them, could that be possible to detect?

Imagine that Charlie is eavesdropping the quantum channel. He receives the information from Alice and then sends it to Bob. That way, neither of them get suspicious that there is a third party involved. Since Charlie cannot copy Alice's qubits due to the no-cloning theorem, what he actually sends to Bob are the results of his measurements (given a specific basis that he chose to measure Alice's qubits). At this time, it is legitimate to ask the question "Would not that change the original message created by Alice?". The answer, is yes. Bob will receive the result of Charlie's measurements, which are not the same as Alice's since she used different bases. Nevertheless, the chances of Bob getting the same bit of Alice is still $50\%$. In other words, Charlie will be only disturbing qubits with probability of $\frac{1}{2}$ each which will negatively affect Bob's chances of agreement with Alice.

At this point, Alice and Bob could easily detect if Charlie or another party was eavesdropping them. To detect the intruder, they have to simply compare the small sequence of bits used with the same basis (the shared secret key). If Alice compares her bit message with Bob's decoded message and they disagree more than a small percentage (due to existence of noise) Charlie's presence will expose as shown in Table 3.4 since it was expected to get the same small-sequence of bits that were measured with the same bases, i.e., if Alice's and Bob's basis are the same, they will very likely get the correct bit of the shared key but, if a third party is involved, the probability of disagreement gets higher. Note that measurement bases used by Alice and Bob are public, but the secret key of $0$'s and $1$'s are still a secret, which means we can use the small sequence of $0$'s and $1$'s as a secret shared key between Alice and Bob.

| Alice's random bits | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|
| Alice's random bases | × | + | + | × | + | × | × | + |
| Information sent by Alice | ↘ | ↑ | → | ↗ | → | ↘ | ↗ | ↑ |
| Charlies's random bases | + | × | + | + | + | + | + | × |
| Charlies's random polarizations | ↑ | ↗ | → | ↑ | → | → | → | ↗ |
| Bob's random bases | + | + | × | × | + | × | + | + |
| Bob's random polarizations | ↑ | → | ↗ | ↘ | → | ↗ | → | → |
| Shared Secret Key | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

Table 3.4: Exposure of Charlie's presence in the quantum channel

With that being said, the BB84 protocol sums up to this: Alice wants to safely communicate with Bob, so, she sends a random secret key to him by polarizing her qubits. Charlie, an eavesdropper, cannot measure and transmit Alice's qubits to Bob without disturbing the qubits' state, which will denounce his presence. This shows the power and the safeness not only from the BB84 protocol but from all protocols created after and from those who are currently being develop with the purpose of safe communications in quantum networks using quantum computers.

## 3.6 Quantum Error Correction

In classical computing, error correction is used to control errors in data over unreliable or noisy communication channels by encoding the message with redundant information. Quantum computers are also susceptible to noisy data and errors. In fact, they are extremely more sensitive to noise when compared to classical computers. In order to identify and fix those errors, we use *quantum error correction*. Quantum error correction [18][17, Chapter 11][31, Lectures 8.1 and 8.3] is used to protect quantum information from noise influence. Quantum noise can occur due to noisy information, faulty quantum gates or incorrect measurements which randomizes qubits and leads to errors [32].

Quantum errors are more complex than classical errors since qubits can be in a superposition state having, therefore, not only bit flip errors as a classical computer do, but also phase flip errors. While bit flip errors consist in bits changing from $0$ to $1$ and vice-versa, phase flips errors occur when a qubit in the state $|-\rangle$ flips to the state $|+\rangle$[5].

As stated before, classical computers employ redundancy for classical error correction, namely, the bit flip. The simplest way to protect a bit against this error is using an error correction code called the *repetition code*. This code consists in making multiple copies of a bit making possible to check, at the end of the channel, which bit differs from the others by taking a majority vote.

Suppose that we want to send a bit through a noisy classical communication channel. While the probability to flip a bit is $p$, $1-p$ represents the probability that a bit is transmitted without getting influenced by noise. This channel is usually called binary symmetric channel in information theory.

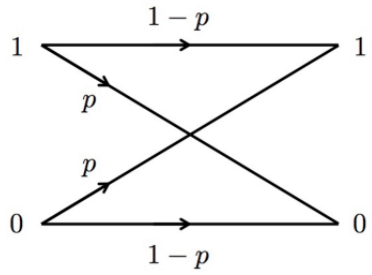



Figure 3.1: Representation of a Binary Symmetric Channel

[5]Although phase flip errors were mentioned, in this section, we will only place a special emphasis on bit flip errors for a more detailed explanation.

As stated before, the best way to protect a bit is by making copies of itself. For that, let us encode

$$\begin{aligned} 0 &\to 000 \\ 1 &\to 111. \end{aligned} \tag{3.35}$$

At this point, we send all three bits through a classical channel and imagine that at the output of that channel we get the bit string $010$. Since $p$ is very small, it is very likely that the middle bit was flipped by the channel, meaning that the bit sent was $0$, i.e., $000$. This type of decoding is usually called majority rule since the decoded output is whatever value appears more times, in this case, $0$. However, since noisy errors are independent, it is possible that two or more bits have flipped instead of one (even if the probability of this event is very small)[6], meaning that the original message could be $111$ instead of $000$. In this case, the decoding will lead us to the wrong result, showing that this type of decoding fails when two or more bits get flipped.

Can we replicate this idea of repetition/redundancy in a quantum algorithm? For instance, imagine that we have a quantum state $|\psi\rangle = a|0\rangle + b|1\rangle$ and we want to send it through a noisy quantum channel. At first look, it might seem appropriate to make three or more copies of $|\psi\rangle$. However, one should not forget that by the no-cloning theorem, it is impossible to copy quantum states. And that is not the only barrier we have in order to add redundancy to the error correction algorithm. There is a continuum of different errors (e.g., bit flip and phase flip) that might occur on a single qubit. Identifying which error occurred seems impossible since it requires an incredible amount of luck to have such precision. Another obstacle that we have here is the fact that if we measure the output's result, the superposition is destroyed. While in classical error correction we could observe the output channel and decide how to decode the message, in quantum error correction the act of observing/measure the information sent destroys the superposition.

Although a straightforward application of classical methods to develop a quantum error correction algorithm is not possible, it is still possible to overcome these problems and create a bit flip correction code.

Suppose that we encode, by linearity, the single qubit state $|\psi\rangle$ in three qubits (that is: $|\psi\rangle = a|000\rangle + b|111\rangle$) as depicted in Figure 3.2.

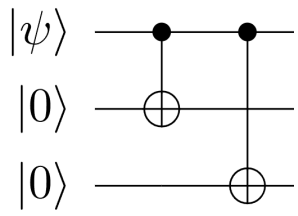



Figure 3.2: Encoding state $|\psi\rangle$ with two more qubits

For example, when using quantum states, if the second qubit is flipped, we get the state $|\psi'\rangle = a|010\rangle + b|101\rangle$. Now, is true we do not want to look at these three qubits and say "The second qubit is $1$

[6]The probability of having one error in one of the three bits is $3p$ and having two errors is $2p^2$.

and the other two are $0$ therefore the error must be in the second qubit" because that act of measurement would destroy superposition. But if we manage to find a way to prove that a certain qubit is different from the other two without measuring it, it would solve the problem. And the solution for this problem is called *syndrome measurement* [33]. This type of measurement provides only information about the error, more concretely, if it happened or not, and it does not give any information about the state of the qubit. In other words, the state $|\psi'\rangle$ remains the same after the syndrome measurement. With this measurement, we are able to identify which bit has flipped, correct it and recover the initial state.

So, after encoding the state, the qubits go through the noisy channel. At this point, four different outcomes can happen:

(i) No error occurred and nothing happens ($I \otimes I \otimes I$);

(ii) The first qubit gets flipped: the state is "moved" to an orthogonal subspace which is spawned by vectors $|100\rangle$ and $|011\rangle$ ($X \otimes I \otimes I$);

(iii) The second qubit gets flipped: the goes to a different orthogonal subspace which is spawned by vectors $|010\rangle$ and $|101\rangle$ ($I \otimes X \otimes I$);

(iv) The third qubits gets flipped to another subspace by vectors $|001\rangle$ and $|110\rangle$ ($I \otimes I \otimes X$);

where $X$ represents the NOT operation.

Then, the qubits arrive to their destination where will be decoded. Before decoding them, we need to know which error happened so we can reverse the bit flip, i.e., if the second outcome happens then we bit flip the first qubit to correct the error and recover the original state. In order to find which error happened a syndrome measurement is performed.

In Figure 3.3 it is represented the complete quantum circuit of the quantum error correction for bit flips.

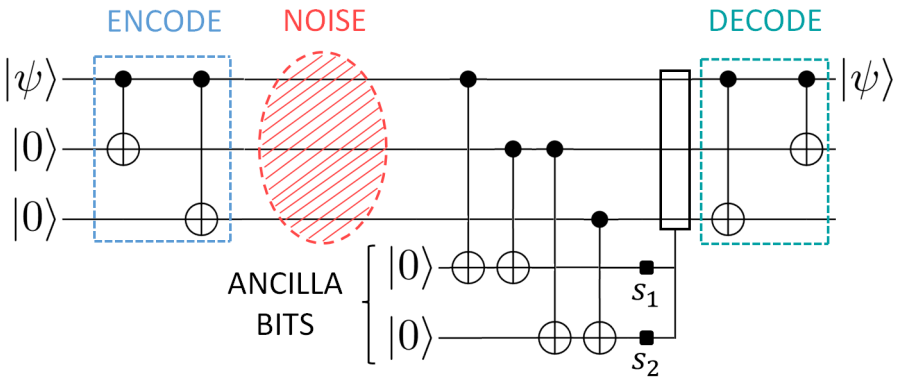



Figure 3.3: Quantum circuit of quantum error correction for bit flips

The syndrome measurement is done with an addition of two extra auxiliary bits called the *ancilla* bits. These bits are both in state $|0\rangle$ so we can perform the syndrome measurement. First, we take the first two qubits of the state has our control qubits and the first *ancilla* bit as our target and apply two CNOT

gates. If the result of this operation ($s_1$) is $0$, both the first and second qubits are in state $0$ or in state $1$. By the same line of argument, we can check the parity of the second and the third qubit resulting in $s_2$.

By knowing the values of $s_1$ and $s_2$ we can identify which error has occurred. Table 3.5 describes the outcomes possible: when $s_1 = 0$ and $s_2 = 0$ nothing happens. If $s_1 = 1$ and $s_2 = 0$ we get to know that it was the first qubit that got flipped. For $s_1 = 0$ and $s_2 = 1$ the second qubit is the one that got flipped and for $s_1 = 1$ and $s_2 = 1$ the third qubit. At this point we know which error occurred, so we can easily correct it and get back the original state.

| $s_1$ | $s_2$ | Bit flip error |
|---|---|---|
| 0 | 0 | $I \otimes I \otimes I$ |
| 0 | 1 | $I \otimes I \otimes X$ |
| 1 | 0 | $X \otimes I \otimes I$ |
| 1 | 1 | $I \otimes X \otimes I$ |

Table 3.5: Possible outcomes depending on the values of $s_1$ and $s_2$

Single bit-flip errors can be corrected with this procedure. Nevertheless, quantum errors tend to be more complex like two or more bits being flipped at the same time, phase flip or a combination of bit-flip and phase-flip. Over the years various quantum codes were created to battle these adversities. The most notorious of these codes is the Shor code, which is able to correct arbitrary single-qubit errors, i.e., bit-flips and phase-flips [31, Lecture 8.5]. To sum up, quantum error correction proves to be an essential tool to protect quantum information in quantum computers by allowing good communications through noisy quantum channels.

## 3.7 Simon's Algorithm

In 1994 Daniel R. Simon presented a solution for a computational problem usually called by the Simon's problem. This algorithm showed that while a classical computer could present a solution in exponential time, a quantum computer could do it in polynomial time, which is way faster. Although this algorithm is not so useful in practice, it demonstrates the key idea for exponential speedups used in posterior algorithms, namely, the Shor's algorithm.

Simon's algorithm [28, Section 6.3] consists in finding a pattern of a given function, i.e., the period. Let $f$ be a function that takes $n$ bits as input and $n$ bits as output

$$f : \{0,1\}^n \to \{0,1\}^n \tag{3.36}$$

such that

$$f(\mathbf{x}) = f(\mathbf{y}) \quad \text{if and only if} \quad \mathbf{x} = \mathbf{y} \oplus \mathbf{s}, \forall \mathbf{x}, \mathbf{y} \in \{0,1\}^n, \tag{3.37}$$

for some unknown $n$-bit string $s \in \{0,1\}^n$. Note that $f$ is a two-to-one function when $s \neq 0$ since it

produces the same output for two different inputs. If $s = 0$, $f$ is a one-to-one function. Notice that

$$\mathbf{x} = \mathbf{y} \oplus \mathbf{s} \Leftrightarrow \begin{cases} \mathbf{x} \oplus \mathbf{y} = (\mathbf{y} \oplus \mathbf{s}) \oplus \mathbf{y} = \mathbf{y} \oplus \mathbf{y} \oplus \mathbf{s} = \mathbf{0} \oplus \mathbf{s} = \mathbf{s} \\ \mathbf{x} \oplus \mathbf{s} = (\mathbf{y} \oplus \mathbf{s}) \oplus \mathbf{s} = \mathbf{y}, \end{cases} \tag{3.38}$$

which means that

$$f(\mathbf{x}) = f(\mathbf{x} \oplus \mathbf{s}), \quad \forall \mathbf{x} \in \{0,1\}^n. \tag{3.39}$$

Simon's problem consists in finding the value of $s$ by repeating $f$ as few times as possible.

For the following example let us suppose that $n = 3$ and $s = 100$. The values of $y$ are therefore given by Table 3.6.

| $\mathbf{x} \in \{0,1\}^3$ | $\mathbf{y} \in \{0,1\}^3$ | $f(\mathbf{x}) \in \{0,1\}^3$ |
|---|---|---|
| 000 | 100 | 010 |
| 001 | 101 | 101 |
| 010 | 110 | 000 |
| 011 | 111 | 110 |
| 100 | 000 | 010 |
| 101 | 001 | 101 |
| 110 | 010 | 000 |
| 111 | 011 | 110 |

Table 3.6: Values of $x$, $y$ and $f(x)$ for $n = 3$ and $s = 100$

As we can easily verify, there are exactly two inputs $x$ that produce the same output $f(x)$. Note that property (3.37) holds: $f(000) = f(100)$, $f(001) = f(101)$, $f(010) = f(110)$ and $f(011) = f(111)$.

This example was for $3$ bits but how exactly should we proceed to solve this problem with $n$ bits?

### 3.7.1 Classical Approach for Simon's Problem

To solve this problem using a classical computer [34], we start by giving different inputs to function $f$ and then observe the corresponding outputs to check which ones are equal and what inputs they respectively correspond. How many different inputs do we have to try in order to find two inputs with same output?

Since $f$ has length $n$, there are in total $2^n$ bit strings. Thus, we can solve Simon's problem in $2^{n-1}+1$ tries at most since half the inputs, $2^{n-1}$, produce unique outputs. Is it possible optimize these results, i.e., use less tries? The answer is yes, but not significantly better.

As a consequence of the birthday paradox[7], the number of tries, that is, the complexity of the algorithm will be reduced to $O(2^{\frac{n}{2}})$, which means that this optimal solution would still need exponential time to solve the problem.

[7] Detailed information on Appendix D

### 3.7.2 Quantum Approach for Simon's Problem

When using a quantum computer to solve Simon's problem, we verify an exponential improvement compared to the previous method. The algorithm can be represented in the following circuit diagram.

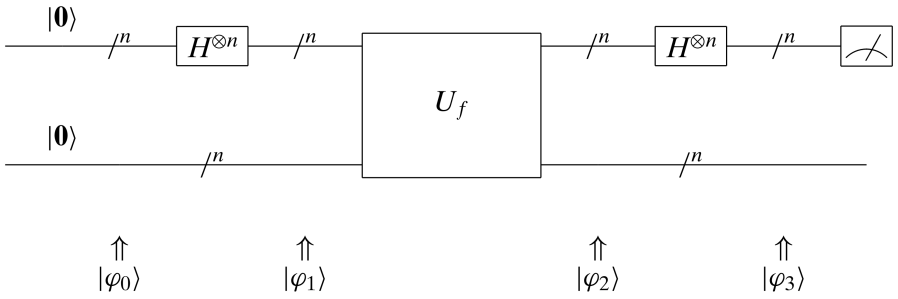



Figure 3.4: Implementation of Simon's algorithm in a quantum circuit

The operation shown in this circuit must be solved several times. However, in contrast with classical methods, in a quantum computer, Simon's algorithm has a number of trials that is linear in $n$.

Note that the circuit is separated in four steps: $|\varphi_0\rangle$, $|\varphi_1\rangle$, $|\varphi_2\rangle$ and $|\varphi_3\rangle$.

We start by having two registers on the input with $n$ qubits each in the $|0\rangle$ state. At the first stage we have

$$\begin{aligned} |\varphi_0\rangle &= |0\rangle^{\otimes n} \otimes |0\rangle^{\otimes n} \\ &= |0\rangle^{\otimes n}|0\rangle^{\otimes n}. \end{aligned} \tag{3.40}$$

At stage two, a Hadarmad gate is applied on the first register, creating an equal superposition of all possible strings of length $n$

$$\begin{aligned} |\varphi_1\rangle &= \left(H^{\otimes n}|0\rangle^{\otimes n}\rangle\right)|0\rangle^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle|0\rangle^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{N-1} |\mathbf{x}, 0\rangle, \end{aligned} \tag{3.41}$$

with $N = 2^n$.

Then we use an oracle $U_f$ to compute the function $f$ on the transformed input. Generally, the oracle has two inputs and two outputs where one output is equal to the first input entry and the second output is the module two addition of the second input entry and the function $f$ applied to the first input entry. In our case, the second input entry is equal to $|0^n\rangle$ which means the result of the second output of the oracle is $f(x)$.

$$\begin{aligned} |\varphi_2\rangle &= U_f \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle|0\rangle^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{N-1} |\mathbf{x}\rangle(|0 \oplus f(\mathbf{x})\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{N-1} |\mathbf{x}\rangle|f(\mathbf{x})\rangle \end{aligned} \tag{3.42}$$

In the final stage, we simply apply a Hadamard gate on the first register again. Note that this time the first register will not be a superposition of all possible inputs, but instead, will be a superposition of all inputs that can produce $f(x)$. Since $f$ is a two-to-one function, the first register will be a superposition of $x$ and $x \oplus s$.

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}\in\{0,1\}^n} \left(H^{\otimes n}|\mathbf{x}\rangle\right) |f(\mathbf{x})\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{\mathbf{x}=0}^{N-1} \left( \left( \frac{1}{\sqrt{2^n}} \sum_{z=0}^{N-1} (-1)^{x\odot z} |z\rangle \right) \otimes |f(\mathbf{x})\rangle \right) \\ &= \frac{1}{2^n} \sum_{\mathbf{x}=0}^{N-1} \sum_{z=0}^{N-1} (-1)^{x\odot z} |z\rangle \otimes |f(\mathbf{x})\rangle \end{aligned} \tag{3.43}$$

After going through all four stages, we perform a measurement at the end of the quantum circuit. The result of this measurement is equally likely to indicate all the $N$ different values of the two-to-one function with each of these values appearing twice ($|z_0\rangle$ and $|z_0 + s\rangle$).

$$\mathrm{H}^{\otimes n} \left( \frac{|z_0\rangle + |z_0 + s\rangle}{\sqrt{2}} \right) = \frac{1}{\sqrt{2^{n+1}}} \sum_{z=0}^{N-1} \left[ (-1)^{z_0\odot z} + (-1)^{(z_0\oplus s)\odot z} \right] |z\rangle \tag{3.44}$$

However, since

$$(z_0 \oplus s) \odot z = (z_0 \odot z) \oplus (s \odot z), \tag{3.45}$$

we get

$$(-1)^{(z_0\oplus s)\odot z} = (-1)^{z_0\odot z}(-1)^{s\odot z}. \tag{3.46}$$

Substituting this last result in (3.44),

$$\left[ (-1)^{z_0\odot z} + (-1)^{(z_0\oplus s)\odot z} \right] = \left[ (-1)^{z_0\odot z} + (-1)^{z_0\odot z}(-1)^{s\odot z} \right] = (-1)^{z_0\odot z} \left[ 1 + (-1)^{s\odot z} \right]. \tag{3.47}$$

Given (3.47), there are two possible alternatives for the value of $1 + (-1)^{s\odot z}$

$$\begin{aligned} s \odot z = 0 &\Rightarrow \left[ 1 + (-1)^{s\odot z} \right] = 2, \\ s \odot z \neq 0 &\Rightarrow \left[ 1 + (-1)^{s\odot z} \right] = 0. \end{aligned} \tag{3.48}$$

This means that, when we measure the final state of the input register, we get, with equal probability, any of the values of $z$ for which one has

$$z \odot s = 0. \tag{3.49}$$

What expression (3.49) implies is that by running Simon's circuit, a bit-string $z$ is obtained such that its inner product modulo-2 with the secret string $s$ is $0$. It is true that we do not obtain the value of $s$ as pretended, but now we do have means to calculate it. Since Simon's algorithm is executed a finite number of times, like $m$ times, we obtain $m$ bit strings

$$z_1 \cdot s = 0, z_2 \cdot s = 0, \ldots, z_m \cdot s = 0 \tag{3.50}$$

which can be solved by using, for example, Gaussian elimination. By doing this we find the value $s$ thus solving Simon's problem. In other words, if we do enough measurements, i.e., if we repeat Simon's algorithm enough times, it is possible to find $s$. The complexity of this result is $O(n)$ which means we get a solution in linear time.

Simon's Algorithm shows that it is possible to have an exponential speedup in solving Simon's problem using a quantum computer instead of a classical one. The power of a quantum computer resides in the quantum parallelism which is what makes possible to perform a large number of operations in parallel and therefore present a solution in polynomial time. Although David Deutsch (1953-) and Richard Jozsa (1953-) were the first to demonstrate that exists a great gap between quantum and classical algorithms [35], Simon's Algorithm served as an inspiration for one of the most important algorithms in the history of quantum computing: Shor's Algorithm.

### 3.7.3 Simon's algorithm: A Numerical Example

In order to better understand Simon's algorithm, let us see a numerical example given Table 3.6 for $n = 3$ bits.

$$\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\} \tag{3.51}$$

$$f\left(\{0,1\}^3\right) = \{000, 010, 101, 110\} \tag{3.52}$$

The two-to-one nature of the function is illustrated below in Figure 3.5.

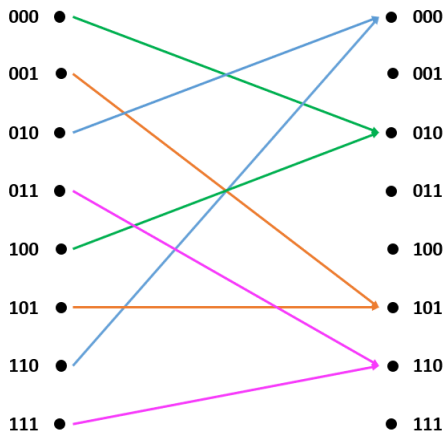



Figure 3.5: Two-to-one function association

The objective is to find the value of $s$ given the mapping depicted in Figure 3.5. First, let us go through the stages of the algorithm according to Figure 3.4.

$$\begin{aligned} |\varphi_0\rangle &= |0^3\rangle \otimes |0^3\rangle \\ &= |000\rangle|000\rangle \end{aligned} \tag{3.53}$$

$$
\begin{aligned}
|\varphi_1\rangle &= \left(H^{\otimes 3}|000\rangle\right) \otimes |000\rangle \\
&= \frac{1}{\sqrt{8}} \sum_{\mathbf{x}=0}^{7} |x\rangle \otimes |000\rangle \\
&= \frac{1}{\sqrt{8}}(|000\rangle \otimes |000\rangle + |001\rangle \otimes |000\rangle + |010\rangle \otimes |000\rangle + |011\rangle \otimes |000\rangle \\
&\quad + |100\rangle \otimes |000\rangle + |101\rangle \otimes |000\rangle + |110\rangle \otimes |000\rangle + |111\rangle \otimes |000\rangle)
\end{aligned}
\tag{3.54}
$$

$$
\begin{aligned}
|\varphi_2\rangle &= U_f\,|\varphi_1\rangle \\
&= \frac{1}{\sqrt{8}} \sum_{\mathbf{x}=0}^{7} |x\rangle |f(x)\rangle \\
&= \frac{1}{\sqrt{8}}(|000\rangle \otimes |010\rangle + |001\rangle \otimes |101\rangle + |010\rangle \otimes |000\rangle + |011\rangle \otimes |110\rangle \\
&\quad + |100\rangle \otimes |010\rangle + |101\rangle \otimes |101\rangle + |110\rangle \otimes |000\rangle + |111\rangle \otimes |110\rangle)
\end{aligned}
\tag{3.55}
$$

$$
\begin{aligned}
|\varphi_3\rangle &= \frac{1}{2^n} \sum_{\mathbf{x}=0}^{N-1} \sum_{z=0}^{N-1} (-1)^{x \odot z} |z\rangle \otimes |f(\mathbf{x})\rangle \\
&= \frac{1}{8} \sum_{z \in \{0,1\}^3} |z\rangle \otimes |\mathbb{F}(z)\rangle
\end{aligned}
\tag{3.56}
$$

Since there is a consider number of calculations to be made in order to solve this equation, it should be a good idea solving it by parts. For this reason, the equality on (3.57) was made.

$$
|\mathbb{F}(z)\rangle = \sum_{\mathbf{x}=0}^{N-1} (-1)^{\mathbf{x} \odot z} |f(\mathbf{x})\rangle \tag{3.57}
$$

Accordingly, one gets

$$
\begin{aligned}
|\varphi_3\rangle =& \frac{1}{8}(|000\rangle \otimes |\mathbb{F}(000)\rangle + |001\rangle \otimes |\mathbb{F}(001)\rangle + |010\rangle \otimes |\mathbb{F}(010)\rangle + |011\rangle \otimes |\mathbb{F}(011)\rangle) \\
&+ \frac{1}{8}(|100\rangle \otimes |\mathbb{F}(100)\rangle + |101\rangle \otimes |\mathbb{F}(101)\rangle + |110\rangle \otimes |\mathbb{F}(110)\rangle + |111\rangle \otimes |\mathbb{F}(111)\rangle)
\end{aligned}
\tag{3.58}
$$

The values of $|\mathbb{F}(z)\rangle$ are given by

$$
\begin{aligned}
|\mathbb{F}(z)\rangle = &(-1)^{z \odot (000)} |f(000)\rangle + (-1)^{z \odot (001)} |f(001)\rangle + (-1)^{z \odot (010)} |f(010)\rangle + (-1)^{z \odot (011)} |f(011)\rangle \\
&+ (-1)^{z \odot (100)} |f(100)\rangle + (-1)^{z \odot (101)} |f(101)\rangle + (-1)^{z \odot (110)} |f(110)\rangle + (-1)^{z \odot (111)} |f(111)\rangle
\end{aligned}
\tag{3.59}
$$

and according to Figure 3.5,

$$
\begin{aligned}
|\mathbb{F}(z)\rangle = &(-1)^{z \odot (000)} |010\rangle + (-1)^{z \odot (001)} |101\rangle + (-1)^{z \odot (010)} |000\rangle + (-1)^{z \odot (011)} |110\rangle \\
&+ (-1)^{z \odot (100)} |010\rangle + (-1)^{z \odot (101)} |101\rangle + (-1)^{z \odot (110)} |000\rangle + (-1)^{z \odot (111)} |110\rangle
\end{aligned}
\tag{3.60}
$$

we get[8]

$$
\begin{aligned}
|\mathbb{F}(000)\rangle = &(+1)|010\rangle + (+1)|101\rangle + (+1)|000\rangle + (+1)|110\rangle \\
&+ (+1)|010\rangle + (+1)|101\rangle + (+1)|000\rangle + (+1)|110\rangle
\end{aligned}
\tag{3.61}
$$

[8] Remember equation (2.34) from Chapter 2

$$|\mathbb{F}(001)\rangle = (+1)|010\rangle + (-1)|101\rangle + (+1)|000\rangle + (-1)|110\rangle + (+1)|010\rangle + (-1)|101\rangle + (+1)|000\rangle + (-1)|110\rangle \tag{3.62}$$

$$|\mathbb{F}(010)\rangle = (+1)|010\rangle + (+1)|101\rangle + (-1)|000\rangle + (-1)|110\rangle + (+1)|010\rangle + (+1)|101\rangle + (-1)|000\rangle + (-1)|110\rangle \tag{3.63}$$

$$|\mathbb{F}(011)\rangle = (+1)|010\rangle + (-1)|101\rangle + (-1)|000\rangle + (+1)|110\rangle + (+1)|010\rangle + (-1)|101\rangle + (-1)|000\rangle + (+1)|110\rangle \tag{3.64}$$

$$|\mathbb{F}(100)\rangle = (+1)|010\rangle + (+1)|101\rangle + (+1)|000\rangle + (+1)|110\rangle + (-1)|010\rangle + (-1)|101\rangle + (-1)|000\rangle + (-1)|110\rangle \tag{3.65}$$

$$|\mathbb{F}(101)\rangle = (+1)|010\rangle + (-1)|101\rangle + (+1)|000\rangle + (-1)|110\rangle + (-1)|010\rangle + (+1)|101\rangle + (-1)|000\rangle + (+1)|110\rangle \tag{3.66}$$

$$|\mathbb{F}(110)\rangle = (+1)|010\rangle + (+1)|101\rangle + (-1)|000\rangle + (-1)|110\rangle + (-1)|010\rangle + (-1)|101\rangle + (+1)|000\rangle + (+1)|110\rangle \tag{3.67}$$

$$|\mathbb{F}(111)\rangle = (+1)|010\rangle + (-1)|101\rangle + (-1)|000\rangle + (+1)|110\rangle + (-1)|010\rangle + (+1)|101\rangle + (+1)|000\rangle + (-1)|110\rangle \tag{3.68}$$

Now we need to calculate for each bit-string

$$|\mathbb{G}(z)\rangle = |z\rangle \otimes |\mathbb{F}(z)\rangle, \tag{3.69}$$

which gives

$$|\mathbb{G}(000)\rangle = (+1)\,|000\rangle\,|010\rangle + (+1)\,|000\rangle\,|101\rangle + (+1)\,|000\rangle\,|000\rangle + (+1)\,|000\rangle\,|110\rangle + (+1)\,|000\rangle\,|010\rangle + (+1)\,|000\rangle\,|101\rangle + (+1)\,|000\rangle\,|000\rangle + (+1)\,|000\rangle\,|110\rangle \tag{3.70}$$

$$|\mathbb{G}(001)\rangle = (+1)\,|001\rangle\,|010\rangle + (-1)\,|001\rangle\,|101\rangle + (+1)\,|001\rangle\,|000\rangle + (-1)\,|001\rangle\,|110\rangle + (+1)\,|001\rangle\,|010\rangle + (-1)\,|001\rangle\,|101\rangle + (+1)\,|001\rangle\,|000\rangle + (-1)\,|001\rangle\,|110\rangle \tag{3.71}$$

$$|\mathbb{G}(010)\rangle = (+1)\,|010\rangle\,|010\rangle + (+1)\,|010\rangle\,|101\rangle + (-1)\,|010\rangle\,|000\rangle + (-1)\,|010\rangle\,|110\rangle + (+1)\,|010\rangle\,|010\rangle + (+1)\,|010\rangle\,|101\rangle + (-1)\,|010\rangle\,|000\rangle + (-1)\,|010\rangle\,|110\rangle \tag{3.72}$$

$$|\mathbb{G}(011)\rangle = (+1)\,|011\rangle\,|010\rangle + (-1)\,|011\rangle\,|101\rangle + (-1)\,|011\rangle\,|000\rangle + (+1)\,|011\rangle\,|110\rangle + (+1)\,|011\rangle\,|010\rangle + (-1)\,|011\rangle\,|101\rangle + (-1)\,|011\rangle\,|000\rangle + (+1)\,|011\rangle\,|110\rangle \tag{3.73}$$

$$|\mathbb{G}(100)\rangle = (+1)\,|100\rangle\,|010\rangle + (+1)\,|100\rangle\,|101\rangle + (+1)\,|100\rangle\,|000\rangle + (+1)\,|100\rangle\,|110\rangle + (-1)\,|100\rangle\,|010\rangle + (-1)\,|100\rangle\,|101\rangle + (-1)\,|100\rangle\,|000\rangle + (-1)\,|100\rangle\,|110\rangle \tag{3.74}$$

$$|\mathbb{G}(101)\rangle = (+1)\,|101\rangle\,|010\rangle + (-1)\,|101\rangle\,|101\rangle + (+1)\,|101\rangle\,|000\rangle + (-1)\,|101\rangle\,|110\rangle \\ + (-1)\,|101\rangle\,|010\rangle + (+1)\,|101\rangle\,|101\rangle + (-1)\,|101\rangle\,|000\rangle + (+1)\,|101\rangle\,|110\rangle \tag{3.75}$$

$$|\mathbb{G}(110)\rangle = (+1)\,|110\rangle\,|010\rangle + (+1)\,|110\rangle\,|101\rangle + (-1)\,|110\rangle\,|000\rangle + (-1)\,|110\rangle\,|110\rangle \\ + (-1)\,|110\rangle\,|010\rangle + (-1)\,|110\rangle\,|101\rangle + (+1)\,|110\rangle\,|000\rangle + (+1)\,|110\rangle\,|110\rangle \tag{3.76}$$

$$|\mathbb{G}(111)\rangle = (+1)\,|111\rangle\,|010\rangle + (-1)\,|111\rangle\,|101\rangle + (-1)\,|111\rangle\,|000\rangle + (+1)\,|111\rangle\,|110\rangle \\ + (-1)\,|111\rangle\,|010\rangle + (+1)\,|111\rangle\,|101\rangle + (+1)\,|111\rangle\,|000\rangle + (-1)\,|111\rangle\,|110\rangle \tag{3.77}$$

One can easily verify that

$$|\mathbb{G}(100)\rangle = |\mathbb{G}(101)\rangle = |\mathbb{G}(110)\rangle = |\mathbb{G}(111)\rangle = 0 \tag{3.78}$$

and

$$\begin{aligned} |\mathbb{G}(000)\rangle &= (+2)|000\rangle \otimes |010\rangle + (+2)|000\rangle \otimes |101\rangle + (+2)|000\rangle \otimes |000\rangle + (+2)|000\rangle \otimes |110\rangle \\ |\mathbb{G}(001)\rangle &= (+2)|001\rangle \otimes |010\rangle + (-2)|001\rangle \otimes |101\rangle + (+2)|001\rangle \otimes |000\rangle + (-2)|001\rangle \otimes |110\rangle \\ |\mathbb{G}(010)\rangle &= (+2)|010\rangle \otimes |010\rangle + (+2)|010\rangle \otimes |101\rangle + (-2)|010\rangle \otimes |000\rangle + (-2)|010\rangle \otimes |110\rangle \\ |\mathbb{G}(011)\rangle &= (+2)|011\rangle \otimes |010\rangle + (-2)|011\rangle \otimes |101\rangle + (-2)|011\rangle \otimes |000\rangle + (+2)|011\rangle \otimes |110\rangle \end{aligned} \tag{3.79}$$

It is possible to write (3.79) as

$$\begin{aligned} |\mathbb{G}(000)\rangle &= (+2)|000\rangle \otimes (|010\rangle + |101\rangle + |000\rangle + |110\rangle), \\ |\mathbb{G}(001)\rangle &= (+2)|001\rangle \otimes (|010\rangle - |101\rangle + |000\rangle - |110\rangle), \\ |\mathbb{G}(010)\rangle &= (+2)|010\rangle \otimes (|010\rangle + |101\rangle - |000\rangle - |110\rangle), \\ |\mathbb{G}(011)\rangle &= (+2)|011\rangle \otimes (|010\rangle - |101\rangle - |000\rangle + |110\rangle). \end{aligned} \tag{3.80}$$

For which $\varphi_3$ is given by

$$\begin{aligned} |\varphi_3\rangle &= \frac{1}{8}(|\mathbb{G}(000)\rangle + |\mathbb{G}(001)\rangle + |\mathbb{G}(010)\rangle + |\mathbb{G}(011)\rangle) \\ &= \frac{|000\rangle}{4} \otimes (|010\rangle + |101\rangle + |000\rangle + |110\rangle) \\ &+ \frac{|001\rangle}{4} \otimes (|010\rangle - |101\rangle + |000\rangle - |110\rangle) \\ &+ \frac{|010\rangle}{4} \otimes (|010\rangle + |101\rangle - |000\rangle - |110\rangle) \\ &+ \frac{|011\rangle}{4} \otimes (|010\rangle - |101\rangle - |000\rangle + |110\rangle). \end{aligned} \tag{3.81}$$

This means that, when the final state of the input register is measured, we obtain $|z\rangle = |000\rangle$, $|z\rangle = |001\rangle$, $|z\rangle = |010\rangle$ and $|z\rangle = |011\rangle$ with equal probability.

Applying this result according to (3.49) we get

$$z \odot s = 0 \Rightarrow \begin{cases} (000) \odot (s_2 s_1 s_0) = 0 \\ (001) \odot (s_2 s_1 s_0) = s_0 = 0 \\ (010) \odot (s_2 s_1 s_0) = s_1 = 0 \\ (011) \odot (s_2 s_1 s_0) = s_1 \oplus s_0 = 0 \end{cases} \tag{3.82}$$

The problem was reduced to a system of linear equations. From this system one could verify that $s_0 = 0$ and $s_1 = 0$. Naturally, $s_1 \otimes s_0 = 0$. The only thing left to find is $s_2$ which is either equal to $0$ or $1$. However, we know for a fact that this is a two-to-one function, which means $s_2$ cannot be $0$, concluding that $s_2 = 1$ and therefore $s = 100$ as expected according to Table 3.6.

# Chapter 4

# Quantum Computation: A Categorical Representation

Following the events of World War I, in the 1930's of last century, five exceptional French mathematicians: Élie Cartan (1869-1951), Jean Delsarte (1903-1968), Claude Chevalley (1909-1984), Jean Dieudonné (1909-1992) and André Weil (1906-1998) created a fictional character going by the name of Nicolas Bourbaki. This character was the collective pseudonym of these group of mathematicians [36]. A variety of textbooks about how to treat and describe mathematics were published under Bourbaki's name. These textbooks were always guided by Bourbaki's motto: everything in mathematics could be explained based on set Theory (mainly initiated by Georg Cantor (1845-1918)). However, as we will see through this chapter, this theory was then took down from another general theory in mathematics: category theory.

## 4.1 Set Theory

In *set theory* [37] a set is a collection of objects. Every group of objects can be created by some sort of rule, for example, packing objects that share similar properties.

In order to introduce set theory in a superficial way, the following definitions are presented:

- A set $A$ containing the numbers $1$ $2$ and $3$ can be written like

$$A = \{1, 2, 3\}. \tag{4.1}$$

  Symbolically, we can write that an element belongs to set $A$ as $1 \subset A$[1]. For example, the number $4$ does not belong to $A$, i.e., $4 \not\subset A$.

- A set can be a subset of another if all of its elements are also elements of another set as described in Figure 4.1

[1] This is read as: $1$ is contained in $A$

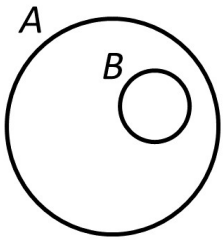



Figure 4.1: ((Sub)set $B$ contained in set $A$

- The empty set is a set which contains no elements and is a subset of any set.
- Two sets may share some objects either by union or intersection of them. While the union of two sets $A$ and $B$ represents the set of all those elements which are either in $A$ or in $B$ ($A \cup B$), the intersection represents only the objects that both sets have in common ($A \cap B$).
- An element of a set may be a set itself and there is a need to distinguish sets and objects. Given a set

$$A = \{\{0\}, \{0, 1\}, \{0, 1, 2\}\}, \tag{4.2}$$

  we can state: the set containing $0$ is an element of $A$ ($\{0\} \in A$). The object $0$ is not an object of $A$ since $A$ just have sets as objects ($0 \notin A$). The set containing the set containing $0$ is a subset of $A$ ($\{\{0\}\} \subseteq A$).
- It is also possible to map a function from $A$ to $B$ trough an arrow. A function is a relation between sets which every element of the domain of $A$ is mapped to the elements of $B$. A function can have two important properties surjective and injective. A function is surjective when all elements of $B$ are reachable. A function is injective when every element of $A$ maps distinct elements of $B$. If both of these properties are observed, we call the function bijective. These properties are depicted in Figure 4.2

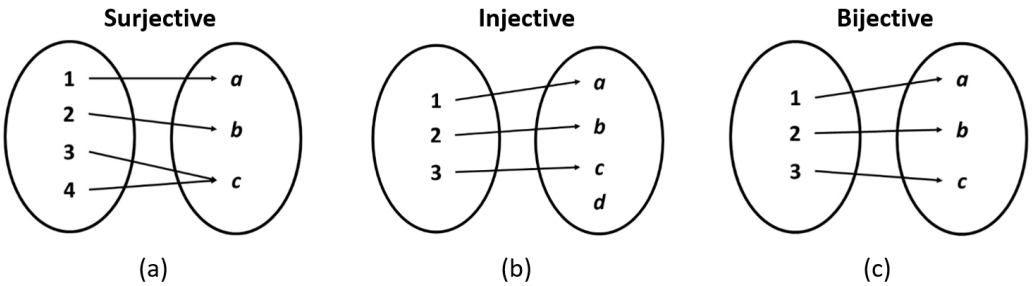



Figure 4.2: From left to right: (a) A non-injective surjective function; (b) An injective non-surjective function; (c) A bijective function (both surjective and injective)

Although set theory seems pretty solid, there are some problems with it. Imagine a set $\Omega$ that contains everything in the universe. If $\Omega$ contains everything, it should contain itself, which leads to an infinite regress of $\Omega$ within $\Omega$. We enter in an infinite loop. This issue is known as Russell's paradox. In a

simpler language the paradox can be described like this: a liar states “I am a liar”. Is he telling the truth or is he lying?

## 4.2 Category Theory

In the 1940’s, two mathematicians named Samuel Eilenberg (1913-1998) and Saunders Mac Lane (1909-2005) were studying algebraic topology when they have found a very abstract way to explain many areas of mathematics. This new way of explaining mathematics was later called as *category theory*, which is usually called in mathematics as abstract nonsense [38]. Although the initial purpose was not to confront set theory and the Zermelo-Fraenkel axiomatic system [2], both Samuel Eilenberg and Saunders Mac Lane defended that category theory was the universal language of mathematics and could solve the problems that set theory could not answer. But what does define a category?

A category is formed by a collection of objects and arrows, which are commonly called as morphisms [39]. A morphism maps two objects: a source and a target. Let us take into account Figure 4.3, if we want to go from a place $A$ to $B$ by applying function $f$ and from $B$ to $C$ by applying function $g$, we can compose this as going from $A$ to $C$ by writing $g \circ f$ where the symbol $\circ$ can be read as ’$g$ after $f$’. In other words, morphisms are a way to go from one object to another. Bear in mind that, contrary to set theory, we cannot ”look” inside the objects, i.e., we simply get to know how they relate to each other’s.

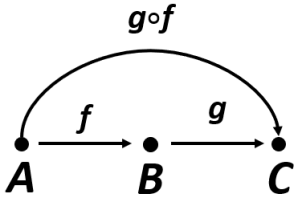



Figure 4.3: Schematic representation of a category with objects $A$, $B$, $C$ and morphisms $f$, $g$, $g \circ f$

In category theory there are two very important axioms that can must always be satisfied so we can consider something as a category: identity and associativity.

The identity property (represented in Figure 4.4) is verified when an element of the set leaves that same set unchanged, that is, a morphism which starts and ends on the same element. It is a neutral operation since it does not change anything like when we add $0$ to a number or multiply any number by $1$.

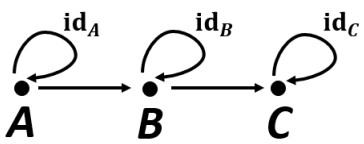



Figure 4.4: Representation of a category with objects $A$, $B$, $C$, morphisms $f$, $g$, $g \circ f$ and the identities $id_A$, $id_B$ and $id_C$

[2] In set theory, Zermelo–Fraenkel axiomatic system (named after Ernst Zermelo (1871-1953) and Abraham Fraenkel (1891-1965)) is a way to formulate a theory of sets, free of paradoxes such as Russell’s paradox.

For the associativity property imagine that we have the following three morphisms, $f$,$g$ and $h$ as depicted in Figure 4.5. The associativity property states that $h \circ (g \circ f) = (h \circ g) \circ f$.

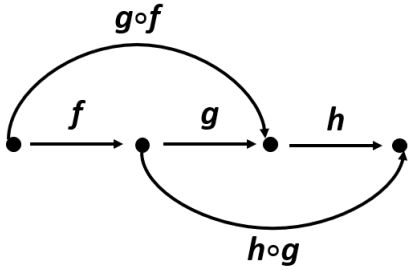



Figure 4.5: Associativity property between three morphisms $f$, $g$ and $h$

In category theory there are many kinds of morphisms. For the following definitions, let us define morphism $f : A \to B$. The morsphism $f$ is called a monomorphism (a generalized concept of injective function) if and only if $f \circ g = f \circ h$ which implies that $g$ = $h$. for all morphisms $g, h : C \to A$. It's dual concept, an epimorphism, is a generalized notion of surjective function. We call a morphism a epimorphism if and only if $g \circ f = h \circ f$ which implies that $g$ = $h$. The morphism $f$ can also be called as an isomorphism when it has an inverse morphism $f^{-1} : B \to A$ such that $f^{-1} \circ f = \mathrm{id}_A$ and $f^{-1} \circ f = \mathrm{id}_B$.

A category can be composed by one single object with many arrows pointing to itself (excluding identity). This type of category is what is usually called as a *monoid*. A monoid can be described as a set equipped with a binary operation (like the sum operation) and a neutral element (which in the case of the sum operation is the number $0$). For instance, is the set of all natural numbers with the sum operation a monoid? The answer is no, because we did not include $0$ but, if we did, it would be a monoid.

We call a *group* to a monoid when every element is invertible. For example, is the set of all real numbers with the operation multiplication a monoid? Yes, since the neutral element $1$ exists in $\mathbb{R}$. But is it a group? No, because $0$ is not invertible.

There are also two important definitions in category theory: *initial object*, an object where there is a unique morphism to any other object including itself (in other words, it is an object where there are only outgoing arrows and just one incoming arrow, the identity) and *terminal object*, where we just have morphisms arriving to that object from all other objects (all arrows converge to this object). Terminal objects can be also called as *singletons* if the set is constituted by just one object.

Commenting about sets with one single object, an empty set is a set with no objects nor morphisms.

Another important notation in category theory is the idea of product and coproduct. Figure 4.6 describes the idea of product. This method applied to computer programming is really reliable when we are fitting data.

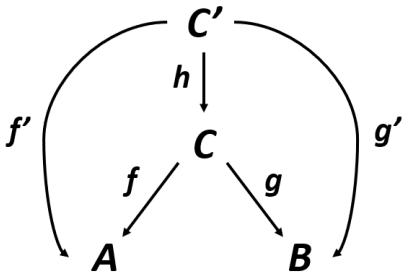



Figure 4.6: Universal property of product

We have represented four objects $A$, $B$, $C$ and $C'$ and five morphisms $f$, $g$, $f'$,$g'$ and $h$. Note that,

$$f \circ h = f' \text{ and } g \circ h = g' \tag{4.3}$$

Object $C$ can fit better data than $C'$ if and only if there is an unique morphism $h$ from $C'$ to $C$ such that (4.3) is satisfied. This can be described in programming language as a factorization of the morphisms $f'$ and $g'$. If (4.3) were a multiplication it would be easy to see as that $f'$ factorizes into $f \times h$ and $f'$ factorizes into $g \times h$. So, this unique common factor $h$ can factorize projection $f'$ and $g'$ (hence the name product).

The coproduct is the dual notion of product, i.e., it reverses the directions of all arrows. Instead of the idea of multiplication, coproduct offers the idea of categorical sum as we represent in Figure 4.7

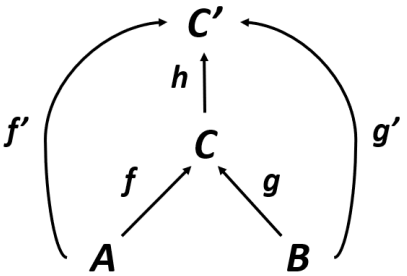



Figure 4.7: Universal property of coproduct

Another interesting notion in category theory are morphisms between categories, that is, *functors*. Now let us take into account Figure 4.8. Assuming two different categories $A$ and $B$ where objects $a, b \in A$ and objects $F_a, F_b \in B$

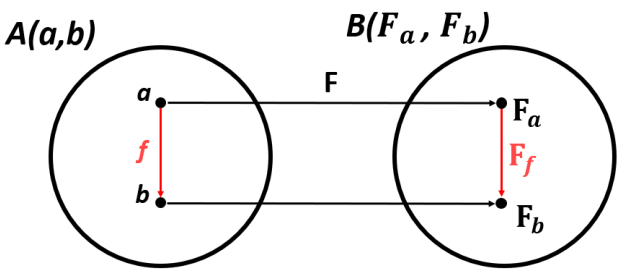



Figure 4.8: Morphism's composition preservation given a functor $F$ with 2 sets $A$ and $B$ and four objects

and given the morphisms $f : a \to F_a$, $f : b \to F_b$, if there is a third morphism $f : a \to b$ in $A$ we must also get a morphism in $B$ such that $F(f) : F(F_a) \to F(F_b)$ where $F$ is a functor. In other words, a functor is defined as a mapping between two sets. Imagine now we have another two objects, one $c \in A$ and another $F_c \in B$ as depicted in Figure 4.9. Note that there is a morphism $g$ from $b$ to $c$ that will be mapped into another from morphism $F_g$ from $F_b$ to $F_c$.

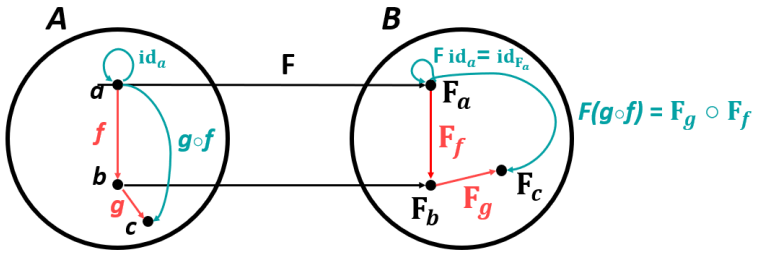



Figure 4.9: Morphism's composition preservation given a functor $F$ with 2 sets $A$ and $B$ and six objects

As we can see, it is possible to compose $a$ and $c$ in $A$ in $g \circ f$. This last morphism will be mapped into an another from $F_a$ to $F_c$, $F(g \circ f)$. On the other hand, we can still compose $F_g \circ F_f$. The result from that composition is simply the morphism $F(g \circ f)$ already mapped. In other words

$$F(g \circ f) = F_g \circ F_f. \tag{4.4}$$

Similarly, since $a$ has an identity morphism $id_a$, that should be mapped into

$$F_{id_a} = id_{Fa}. \tag{4.5}$$

As there are morphisms between categories, there is also morphisms between functors. That phenomenon is called *natural transformations*. Given the functors $F : A \to B$ and $G : A \to B$, a natural transformation associates a morphism $\eta_x : F_x \to G_x$ to every object in $A$, where $x$ can be either $a$ or $b$ such that

$$\eta_b \circ F_f = F_g \circ \eta_a \tag{4.6}$$

as exemplified in Figure 4.10.

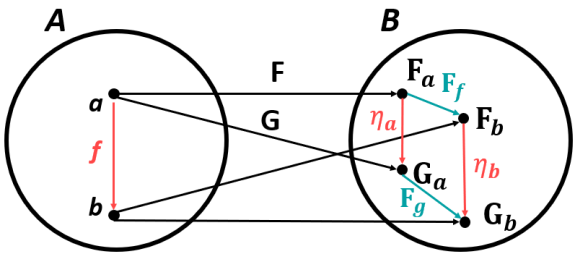



Figure 4.10: Representation of a natural transformation given the functor $F$ and $G$

After viewing some proprieties of both set theory and category theory we can point some essential differences: in set theory there are two crucial concepts, the concept of a set containing an element,

and the concept of function such that there is an application from a set $A$ to a set $B$ where all elements of $A$ have one and only one outgoing arrow that reach $B$. On the other hand, in category theory we cannot look "inside" of the objects, that is, we do not know what they are, their content, but we do know how they can relate to other objects by using morphisms. However, it is hard to define the membership of a certain object in a set due to category theory being an abstract theory. Nevertheless, morphisms between objects provides an uniform language that preserves the mapping of mathematical objects of an arbitrary type, therefore, answering questions that set theory cannot.

Nowadays category theory is used not only in many areas of mathematics but also proved to be extremely useful in some areas of computer science, namely, quantum computation. By conjugating category theory with quantum information, it is possible to simplify quantum circuits by representations purely diagrammatic of quantum processes, which is the main theme of this chapter.

## 4.3 Categorical Quantum Mechanics

### 4.3.1 Diagram Language: An introduction

In order to understand this new notation of categorical quantum mechanics, it is necessary to first introduce some knowledge into diagrammatic equations before presenting the quantum computing part [11, Chapter 2]. Figure 4.11 shows how a function $f$ is graphically represented.


Figure 4.11: Diagrammatically representation of a function $f$

Worth noticing that having a single wire with nothing else involved can be interpreted as applying the identity morphism, which does not do anything.

Figure 4.12 depicts a more interesting and practical example: a function applied between objects.

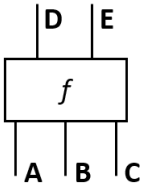



Figure 4.12: Function $f$ outputs $D$ and $E$ given the inputs $A$, $B$ and $C$

In terms of equations, Figure 4.12 can be written as

$$f_{ABC}^{DE} \tag{4.7}$$

Please note that the subscripts appearing below the function will always represent the inputs and the superscripts the outputs, meaning that the direction taken to study and analyze diagrams in this dissertation will be the same as instructed in the book 'Picturing quantum processes A first course in quantum theory and diagrammatic reasoning': from the bottom to the top.

When working with more than one function, the diagram can be represented as the following

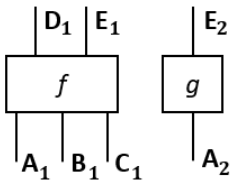



Figure 4.13: Diagrammatically representation of two function $f$ and $g$ in parallel

. With that being said, the written equation can get a bit more robust, yet, trivial.

$$f^{D_1E_1}_{A_1B_1C_1} g^{E_2}_{A_2} \tag{4.8}$$

The numbers under the objects eliminate the ambiguity whenever we have more than one wire with the same object in a single diagram. Also note that when writing the written expression, the order is irrelevant since they represent the same category.

In Figure 4.13 the wires were processed in parallel, i.e., simultaneously. This process is called parallel composition, where two or more diagrams are processed side-by-side. This synchronicity can be represented by the symbol $\otimes$. The composition operation is associative as illustrated in 4.14.

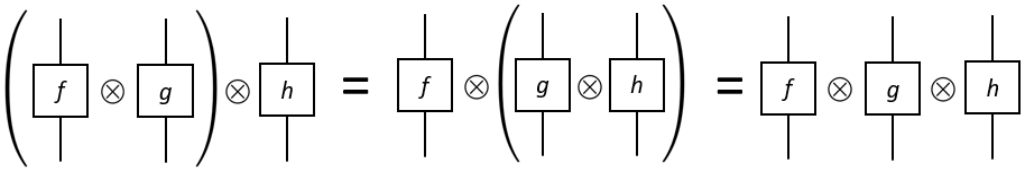



Figure 4.14: Demonstration of associativity in composition operation

Composition also has a unit which is represented by a function composed with an empty diagram (does nothing), which will lead to the same function with no alteration.

Another operation used in quantum picturalism is the sequential operation. The sequential operation is represented by the symbol $\circ$ as represented in Figure 4.15.

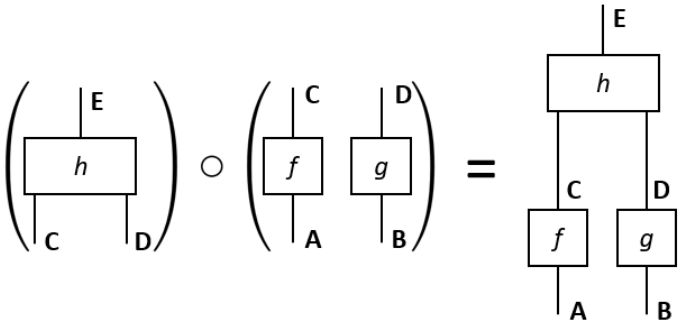



Figure 4.15: Representation of a sequential composition

This operation is also associative and admits the identity property as well (which is a wire and therefore changes nothing)[3].

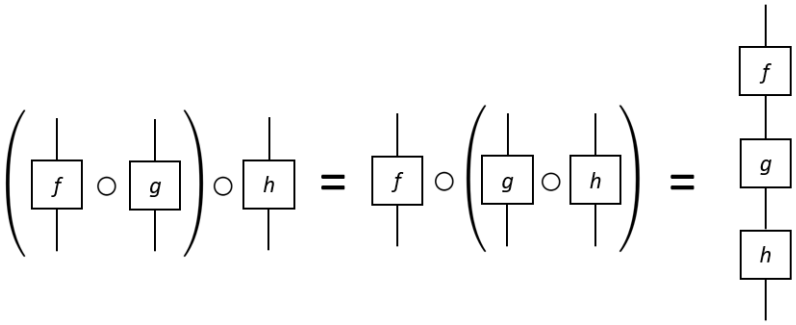



Figure 4.16: Demonstration of associativity in a sequential composition

At this point we only explored a few basics of this new graphical language without applying any relevant inputs or outputs. Figure 4.17 illustrates an initial process usually called *state*. States work as start objects of sets whereby have no inputs.


Figure 4.17: Representation of a state

On the other hand, we have states dual, *effects*. Effects has no outputs for which represents the ending of a process, meaning that it does not exist any more wires after it as represented in Figure 4.18.


Figure 4.18: Representation of an effect

Note that given a certain state, we can produce a certain effect. This relation (Figure 4.19) can be interpreted as a probability.

[3]Numbers work as a multiplication process in this language. They do not have any inputs or outputs since they are constants but the properties of associativity (Figure 4.16)and identity are still valid for which it is not repeated.


Figure 4.19: Composition between a state and an effect

Worth noticing that a state $\psi$ is bipartite if and only if it can be separated into two states $\psi_1$ and $\psi_2$ as follows:

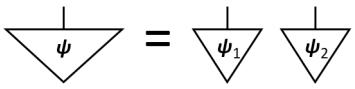



Figure 4.20: Representation of a bipartite state $\psi$

The quantum part of these diagrams comes in the form of quantum maps (in contrast with linear maps that we just introduced). Contrary to linear maps, quantum maps are positive maps which generate complex numbers (useful for the use of probabilities) that can be mixed with quantum states and quantum effects. To distinguish quantum maps from linear maps we simply "cut" the edge of the graphical element as exemplified in Figure 4.21.

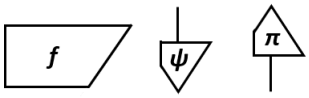



Figure 4.21: Example of quantum elements used in quantum maps

But before diving any further on the quantum side of this language, seven crucial definitions must be established: cups, caps, transpose, trace, unitarity, adjoint and conjugate [11, Chapter 3].

*Cups and caps* can convert processes to states and vice-versa as Figure 4.21 illustrates. As it is easily verified, they are the inverse of each other. They come to be very handy in the construction of quantum diagrams and when dealing with transpose and trace.


Figure 4.22: Diagrammatically representation of cups (on the left) and caps (on the right)

Given a process $f$, its *transpose* can be depicted as

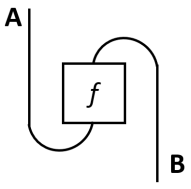



Figure 4.23: Transpose of a process $f$

generating a new process. A transpose can be interpreted as a 180° rotation. The real advantage of this notation comes when transposes interact with cups and caps. In Figure 4.24 we have an example of that. Note that we just slide the box trough the cup from left to right.

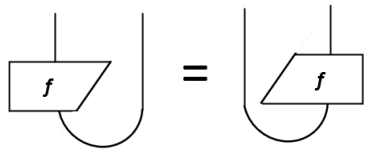



Figure 4.24: Transposing a process $f$ through a cup

When using states and effects, one can clearly see that there is a bijective correspondence, that is, a state is turned into an effect and vice-versa.

A *trace* can be depicted when the input type is the same as the output, i.e.,

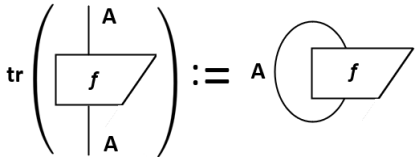



Figure 4.25: Example of how applying a trace in a graphical language

Another property that can significantly simplify the diagram is *unitarity*. A process $f$ is unitary if

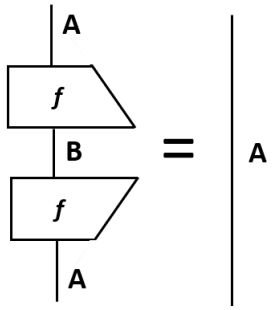



Figure 4.26: Simplification of a diagram through the use of unitarity

Worth noticing that image depicted is also called as an isometry.

An *adjoint* process is simply a vertical reflection of a state as illustrated in Figures 4.27 where † symbolizes the adjoint operation as we have covered in Chapter 2.


Figure 4.27: Representation of the adjoint operation

The *conjugate* of a process is a combination of the adjoint and the transpose (by either being the adjoint of its transpose or the transpose of its adjoint). Graphically speaking, this is represented by a

horizontal reflection as illustrated in Figure 4.28.

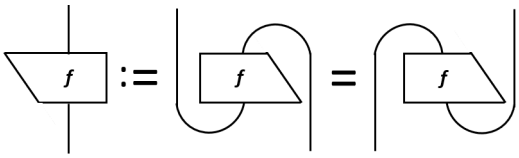



Figure 4.28: Representation of the conjugate of a process $f$

Figure 4.29 summarizes in a simplified way the relations between the transpose, adjoint and conjugate given a process $f$. The same conclusions can be applied when dealing with states, effects and numbers.

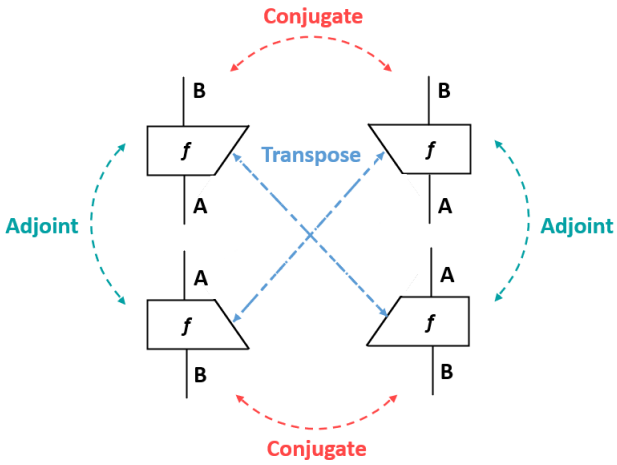



Figure 4.29: Relation between tranpose, adjoint and conjugate of a process $f$

### 4.3.2 Diagram Language: The No-Cloning Theorem

A nice way to start conjugating quantum mechanics and graphical calculus is by showing an example, namely, a known example: the no-cloning theorem.

Before proceeding to the demonstration of the theorem, one should note the following proposition: if a theory is described by string diagrams, and all bipartite states are $\otimes$-separable, then all processes are $\circ$-separable.

Remember that if cloning was possible then we would have the input duplicated at the output. In diagram language that would mean that given a quantum state $\psi$ we would have at the outputs two $\psi$, the original one and the duplicated one. Note that these states are $\otimes$-separable.

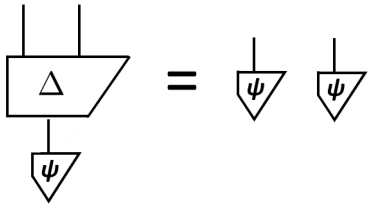



Figure 4.30: Diagrammatically notation of a cloning procedure

There are some proprieties that should be taken into account due to this cloning process. If we are producing two identical copies of a state as an output, it does not matter if we switch them as depicted in Figure 4.31.

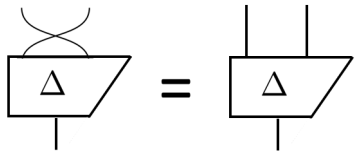



Figure 4.31: Interchanging the outputs of a process $\Delta$

Additionally, if this time we are trying to clone two different states $A$ and $B$, it should be possible to create a state of the type $A \otimes B$ by cloning each system individually.

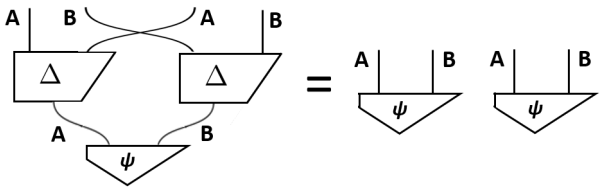



Figure 4.32: Creating state $A \otimes B$ by cloning each system individually

Since we are trying to clone an input of a single type (e.g. type $A$), the outputs will be of that same type as well. Nevertheless, so the reader can be on track with all the deformations and manipulations presented next, we identified each output with a different subscript of type $A$, even if they all represent the same type $A$. Bear in mind that

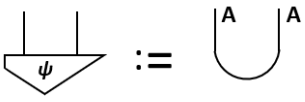



Figure 4.33: Notation for a cloning state of type $A$

Now, let us see if the previous proposition is true when we are trying to clone states. If cloning is possible, then (and according to the equivalence in Figure 4.33 and Figure 4.32)

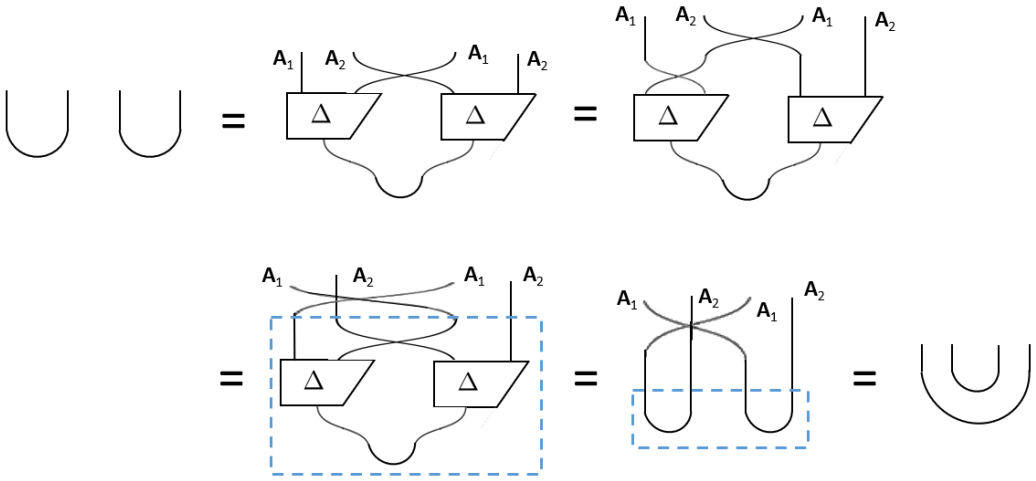



Figure 4.34: Diagrammatically manipulation of two cloned states

Note that for the second equality we simply applied the property described in Figure 4.31, where we interchanged the output since they are the same type: $A$. After applying some graphical manipulation and make use of the property of Figure 4.32 (dashed blue rectangle), it should be easy to understand the last equality obtained by thinking three-dimensionally (simply imagine we are bending the wires with our hands).

Now, converting the external outputs to inputs on the initial expression and on the final deduced expression on Figure 4.34, we get

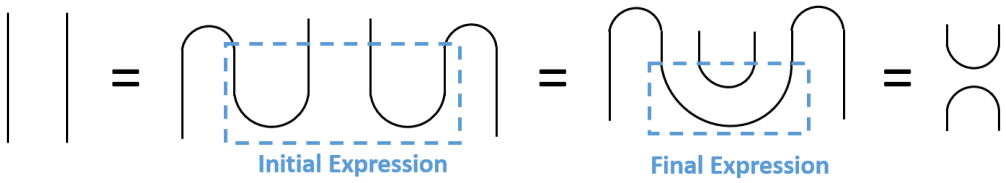



Figure 4.35: Conversion of the external outputs of Figure 4.34 into two different inputs

With the equality obtained on Figure 4.35 in mind, we can conjecture the following equalities:

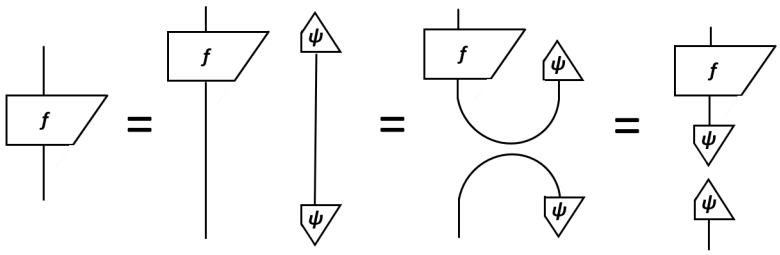



Figure 4.36: Demonstration that any process $f$ is $\circ$-separable

which means that any process $f$ that has an input state of type $A$ is $\circ$-separable, proving the proposition.

The assumption that makes impossible to clone a state is Figure 4.32. In language of string diagrams it is impossible to copy a non-separable state by manipulating each of its subsystems. As we have seen in Chapter 2, when it comes to entangled states, we can know everything about a system and know nothing about its parts, meaning that not all states are bipartite. There exists states that are no-separable namely, the entangled ones. With that being said, the equality presented in Figure 4.32 is not always true (as it is not for this example) for which it is impossible to clone a quantum state, proving once again, the no-cloning theorem. One should note that we could have used Figure 4.37 as our starting point (instead of Figure 4.30) since it resemblances the example of Alice's and Bob's music albums of Chapter 3, where they have a raw material (now represented as state $\phi$) that would transform into the copied state.

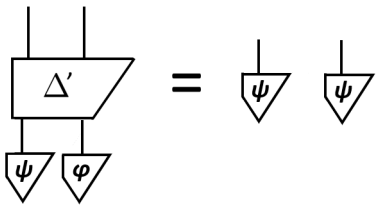



Figure 4.37: Cloning process of a state $\psi$ with the use of raw material $\phi$

For sake of simplicity, we did not went with this approach although would lead us to the same results.

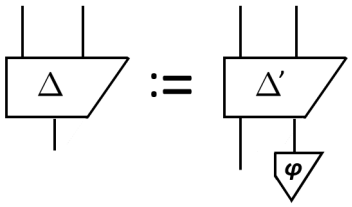



Figure 4.38: Equivalence of processes from Figures 4.30 and 4.37

At this point the reader might be wondering why did we go for all of this when it seems we are just complicating things in the way that we are just showing a different and abstract way of known results seen in previous chapters. The truth is that we do not have the full picture of quantum picturalism yet and, without that, we cannot simplify circuit diagrams and mathematical formulas. One essential ingredient is still missing and that is the *ZX-calculus*.

### 4.3.3 ZX-Calculus

The *ZX-calculus* [11, Section 8.4] [40] [41] is a graphical language for reasoning about linear maps between qubits introduced by Bob Coecke and Ross Duncan in 2008. The graphical representation of these maps is called *ZX-diagrams* which can be related to quantum circuit diagrams but with a different notation as we will see. In 2017 it was proved that any equality between linear maps can be proven diagrammatically with ZX-calculus, meaning that all reasoning about quantum computation can be done with it. Be aware that the following section of ZX-calculus will be exemplified with the diagrams hori-

zontally (in contrast with the diagrams we just introduced for categorical quantum notation) for a more practical and better understanding. Nevertheless, it will only be for the purpose of introducing the topic since further examples will have back the vertical orientation.

Quantum circuits are composed by wires and quantum gates. ZX-diagrams use them as well but not quite with the same representation as we have seen in Chapter 2: there are some translations from quantum circuits to ZX-diagrams. These translations and other important notations are described in Figure 4.39[4].

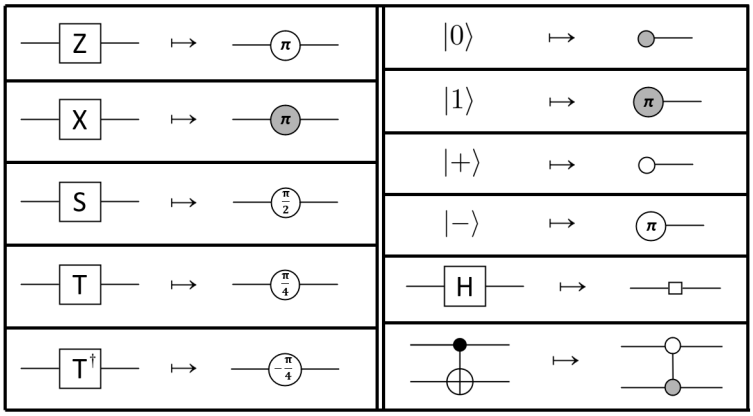



Figure 4.39: Translation table from quantum circuits elements to ZX-diagram elements

One legitimate question is why the name ZX-calculus and why we did not mention $Y$? Remember, $Y$ is a composition of both $X$ and $Z$, meaning that rules relating to the $Y$-eigenbasis can be derived from the equations that we will introduce. With that being said, we can very easily go from a quantum circuit to a ZX-diagram like depicted in Figure 4.40.

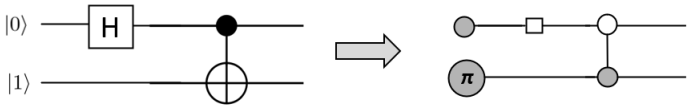



Figure 4.40: Example of a translation from a quantum circuit to a ZX-diagram

Again, one should might ask: why are we complicating things once again when the diagram is exactly like the circuit but with a different representation of the gates? The difference becomes clearly evident when, for example, a $Z$-phase gate commutes through a CNOT gate as illustrated in Figure 4.41.

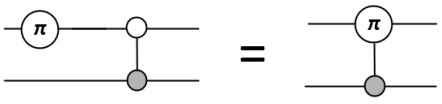


Figure 4.41: ZX-diagram simplification by matching dots of the same colour

When it comes to quantum circuits there is not much flexibility when trying to simplify the circuit. However, when using ZX-diagrams, we can take advantage of one essential property: dots with the

[4]Originally, the white and grey dots were represented by the colours green and red respectively. The change of colour resulted with the objective to reach a much larger public

same colour can commute through each other. In this case, the white dots merged together, simplifying the diagram.

With this great rule in mind and applying other properties, it is possible to simplify quantum circuits even more. For example, if a grey dot has a single wire connected to a white dot then the first one gets copied, deleting the white dot as shown in Figure 4.42.

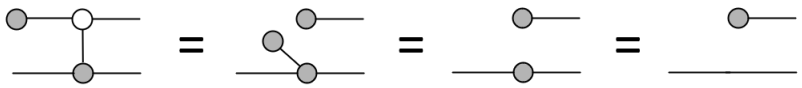


Figure 4.42: Simplification of a ZX-diagram through the copying dots

Keep in mind that in the last equality we have turned a grey dot into a simple wire with no dots. This is due to another rule which states that dots (grey or white) with just two wires connected to them can be turned into a single wire with no dots.

It should come to notice that it is possible to generate quantum entanglement by using ZX-diagrams. One raw translation of it would be going directly from the quantum circuit that generates entangled Bell's states to a diagram as showed in Figure 4.40. Nevertheless, we now know how to simplify this last ZX-diagram which in fact turns simply into a cup[5] if the inputs are both $|0\rangle$[6].

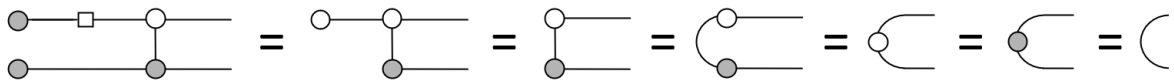


Figure 4.43: ZX-diagram representation of the Bell state $|\Phi^+\rangle$

The Hadamard gate takes a fundamental place in quantum circuits, so it takes in ZX-diagrams. Hadamard gates have two important properties in ZX-diagrams: one being self-inverse, that is,

Figure 4.44: Self-inverse property of the Hadamard gate in ZX-diagrams

and another property which is the conjugation between the Hadamard gate and the Pauli gates $X$ and $Z$ (expression (2.32)).

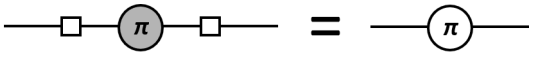



Figure 4.45: Diagrammatically transformation from a gate $X$ to a gate $Z$

One can generalize this property as depicted in Figure 4.46

[5] Remember that in this subsection we are drawing diagrams horizontally.
[6] Take into account the last equality from Figure 4.42.

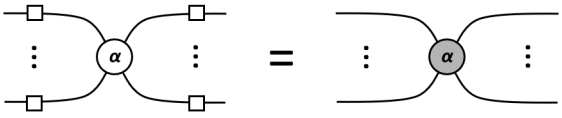



Figure 4.46: Generalization of the second Hadamard property

or manipulate Hadamard gates in various ways as described in Figure 4.47.

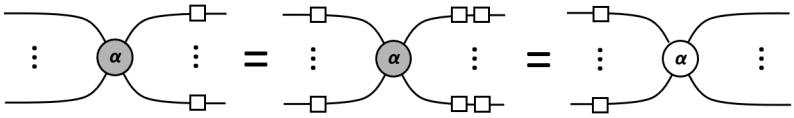



Figure 4.47: Manipulation of a diagram using Hadamard gates

Now that we have presented the fundamental tools of ZX-calculus, let us introduce one very important generator in ZX-diagrams: *spiders*.

There are two important types of spiders in this new language: the $Z - spider$ and the $X - spider$. The Z-spider is represented by a white dot with any number of inputs and outputs as represented in Figure 4.48.

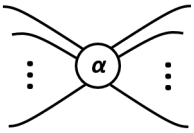



Figure 4.48: Representation of a Z-spider

In terms of linear map, the Z-spider can be interpreted as follows:

$$|0\cdots 0\rangle\langle 0\cdots 0| + e^{i\alpha}|1\cdots 1\rangle\langle 1\cdots 1| \tag{4.9}$$

Note that through equation (4.9) we obtain the eigenbasis of the Z matrix, $|0\rangle$ and $|1\rangle$.

On the other hand, X-spiders work exactly like Z-spiders excepting for the fact that they are represented by grey dots instead of white (Figure 4.49)

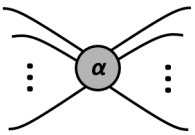



Figure 4.49: Representation of a X-spider

and they are defined with the eigenbasis of the $X$ matrix, $|+\rangle$ and $|-\rangle$.

$$|+\cdots+\rangle\langle+\cdots+| + e^{i\alpha}|-\cdots-\rangle\langle-\cdots-| \tag{4.10}$$

For example, if a X-spider has just one input and one output and a phase $\alpha$ we get

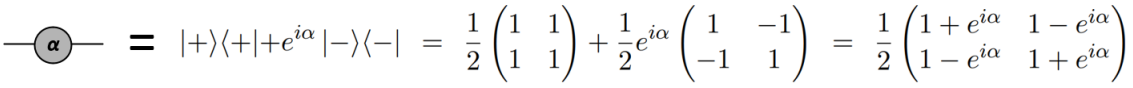



Figure 4.50: Matrix representation for a X-spider with one input and one output and a phase $\alpha$

The matrix obtained represents a rotation of the Bloch sphere by an angle $\alpha$ over the $X$ axis.

We can also represent the basis states using spiders as described in Figure 4.51.

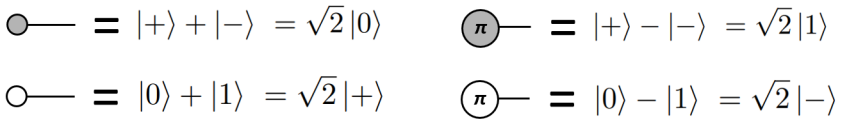



Figure 4.51: Representation of the basis states using spiders

These last results should at least raise an eyebrow to the reader: even though the states are correct, the scalars are different from what we have seen in Chapter 2. In fact, in ZX-diagrams, scalars do not have any significance for which we can ignore them [7].

As we have seen in Figure 4.39, it is possible to represent quantum gates diagrammatically. Even if the translations seem immediate, there is a graphical logic behind it. For instance, let us study the CNOT gate. First, we need a phaseless Z-spider with one input and two outputs. Then we horizontally compose it with a single wire, i.e., the identity.

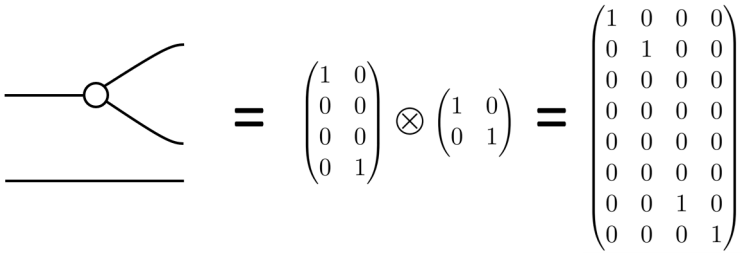



Figure 4.52: Composition of a Z-spider with the identity

Then we compose a wire with a phaseless X-spider as depicted in Figure 4.53.

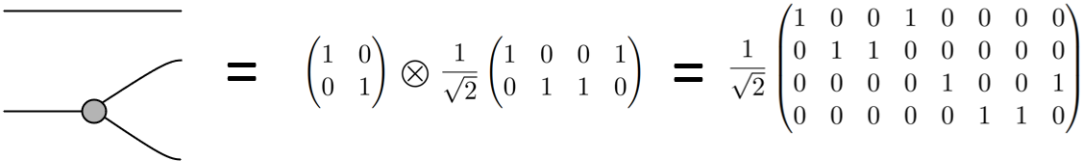



Figure 4.53: Composition of the identity with a X-spider

If we compose the diagrams on Figures 4.52 and 4.53 we get indeed the matrix representation of the CNOT gate.

[7] We can ignore all global non-zero scalar factors

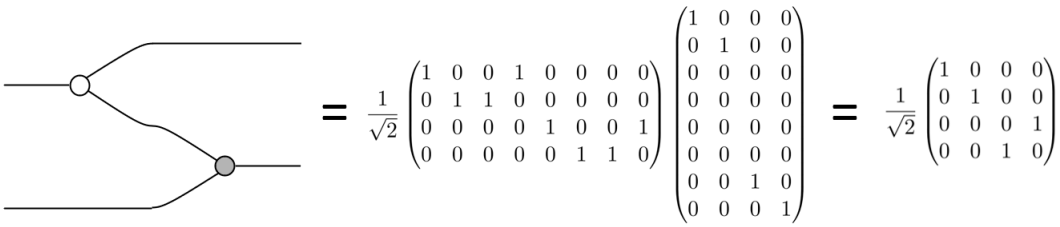


Figure 4.54: Matrix representation of a CNOT gate using spiders

Bear in mind that even if the diagram on Figure 4.54 is not vertically aligned as showed in Figure 4.39, it still represents the CNOT gate. As a matter of fact, one can easily verify that no equality was broken by checking the matrix obtained.

Another property that spiders have is that they can be interpreted as symmetric tensors, i.e, if its wires are swamped, nothing really important changes. Figure 4.55 shows an example of a swap.

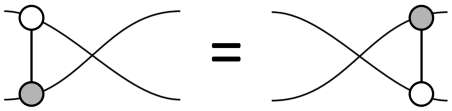


Figure 4.55: Example of swap generator interacting with spiders

Imagine now the scenario where two spiders of the same colour are connected. As seen before, it is possible to simplify the diagram by fusing the spiders (and with that, adding their phases) as demonstrated in Figure 4.56. This adding represents two rotations in the same direction on the Block Sphere.

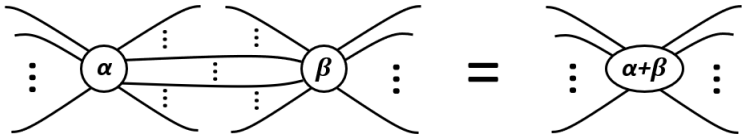



Figure 4.56: Commutation of spiders by adding their phases

One can make use of the identity removal which eliminates the existence of self-loops in spiders as depicted in Figure 4.57.

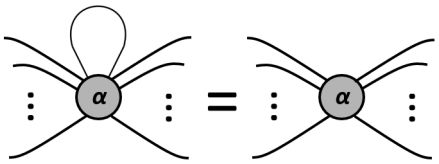



Figure 4.57: Elimination of an identity self-loop

One last rule should be brought to the table. Given a Z-spider, the equality of Figure 4.58 must hold for any process $f$.

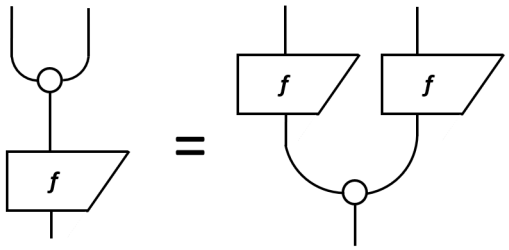



Figure 4.58: Process $f$ commuting through a Z-spider

### 4.3.4 Quantum Oracle Through Diagrams

It all comes down to this. It is time to apply everything discussed in this dissertation. Now we have all the tools to dive into quantum algorithms using this new graphical notation and verify if this new view of categorical quantum mechanics has any advantages over quantum circuits.

Although Simon's algorithm was mentioned in Chapter 3, its graphic notation is outside the scope of this dissertation. Nevertheless, we will categorically present the crucial part of not only Simon's algorithm but also from many other quantum algorithms: the quantum oracle [11, Section 11.2]. Bear in mind that since we are no longer studying the ZX-diagrams we will no longer represent diagrams with horizontal orientation wherefore the following diagrams will be vertically oriented as 'Picturing Quantum Processes: A first course in quantum theory and diagrammatic reasoning' does.

As stated in previous Chapters, an oracle can be interpreted as a black box that executes unitary operations. In terms of linear map, a function $f$ is unitary if exists a bit-string $i$ such that $f(i) = 1$.

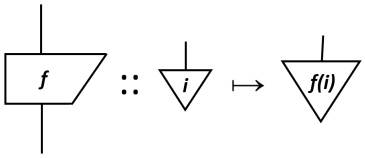



Figure 4.59: Turning $f$ into unitary

However, this linear map (and consequently the quantum map $\widehat{f}$) can only be unitary if $f$ is a bijection from its inputs to its outputs. To accomplish that, we can use some properties already learned from this Chapter.

First and foremost, remember that Simon's Algorithm takes $n$ bits at the input, meaning that we need $n$ spiders to start the oracle's representation as depicted in Figure 4.60.

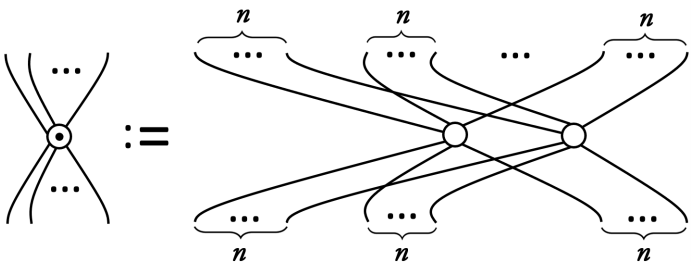



Figure 4.60: Representation of $n$ spiders

A quantum map $\widehat{f}$ is unitary if the white-dot spiders and the grey-dot spider are complementary (i.e., we use a grey-dot spider as an output of the $n$ spiders).

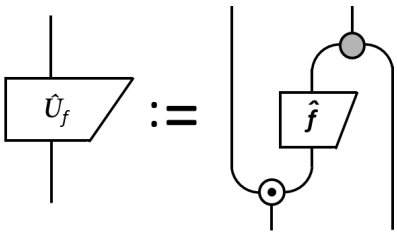



Figure 4.61: Condition for unitarity of a quantum map

To prove it, let us manipulate Figure 4.61. Taking into account the equality from Figure 4.58 and the fact that classical maps are self-conjugate:

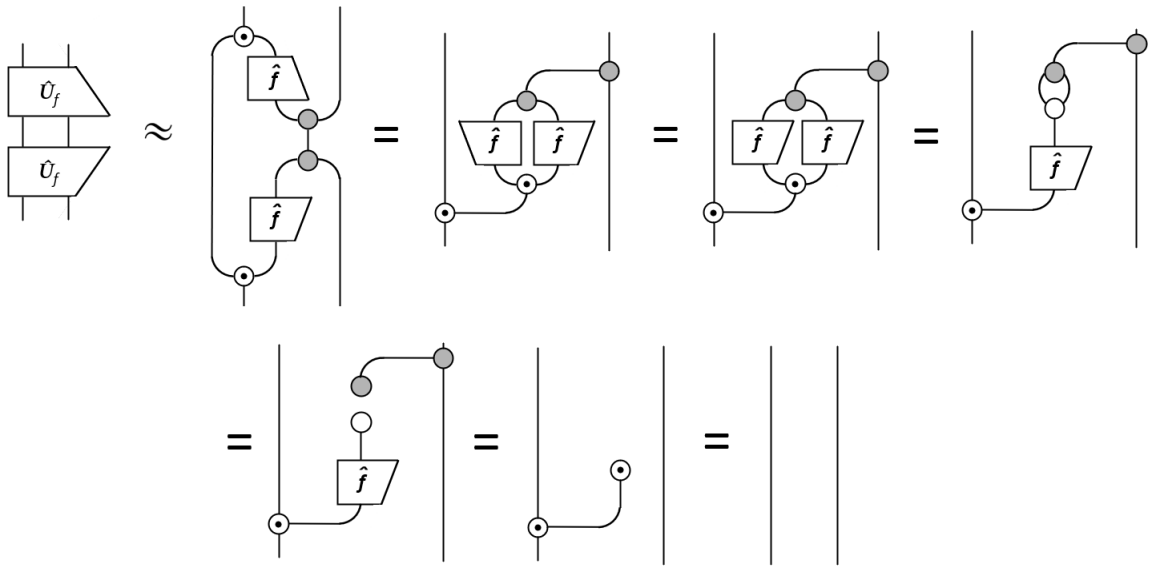



Figure 4.62: Graphical manipulation given two $\widehat{U}_f$

Given this result, one could use it to substitute the quantum map $\widehat{f}$ in Figure 4.61,

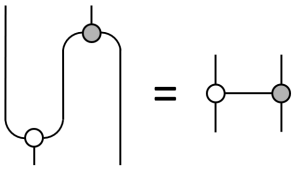


Figure 4.63: Proving unitarity of a quantum map $\widehat{f}$

for which is unitary.

The black-box $\widehat{U}_f$ turns the quantum map $\widehat{f}$ into a unitary by adjoining an extra input matching $\widehat{f}$'s output (going out of the grey dot) and extra outputs matching $\widehat{f}$'s inputs (coming in from the bottom of the white dot). By doing this, we are precisely doing the oracle's operation: the state that comes from the left of $\widehat{U}_f$ gets copied, then the first of these copies goes to the first output of $\widehat{U}_f$ and while this happens, the other copy goes into the input of $\widehat{f}$. The corresponding output of this input gets XOR'ed with the second input of $\widehat{U}_f$.

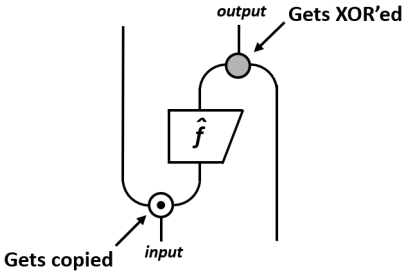



Figure 4.64: Representation of a quantum oracle's operation

Remember that the second input of $\widehat{U}_f$ is $|0\rangle$ in Simon's algorithm. With that being said and given a string of qubits into the first input, we get

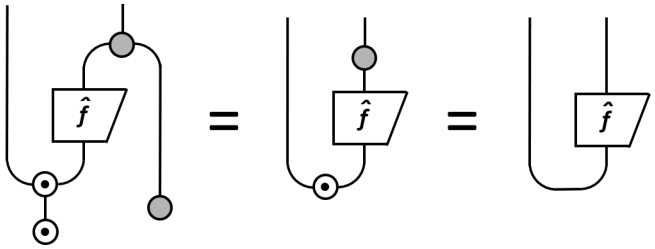


Figure 4.65: Representation of $\widehat{U}_f$ having a string of qubits for the first input and $|0\rangle$ for the second one

Combining this result with the statement of Figure 4.59, we obtain

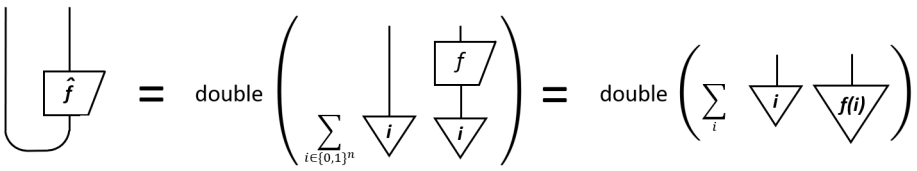



Figure 4.66: Diagrammatically representation of a quantum oracle

This result means that we get the entire function $f$ encoded as a state, i.e., a superposition of the values of $f$ for every input. There is no denying that the diagram representation of a quantum oracle turned to be simple, practical and succinct when compared with the quantum circuit one. And notice that the result obtained on Figure 4.66 describes exactly what an oracle does: a quantum procedure that processes all inputs at the same time.

# Chapter 5

# Conclusions and Forthcoming Research

After visiting all chapters of this dissertation, the following paragraphs give not only a summary of the main conclusions given the subjects of each chapter but also a final remark on quantum computation as well for future prospects.

## 5.1 Conclusions

Chapter 2 gently introduces the fundamentals about quantum computation and quantum mechanics. It starts by defining a qubit, how to represent it on the Bloch Sphere given a certain basis and proving its fundamental properties, namely, quantum entanglement and quantum superposition. Then game based on PR boxes was analyzed with the purpose of exposing two different views about quantum nonlocality: a skeptical one, given by the notorious EPR group and another view coming from the perspective of quantum mechanics. We came to a conclusion that while EPR defended that with the presence of hidden variables, that is, with pre-planned strategies between the players, the chances to win were, at most, 75%. On the other hand, quantum mechanics, by taking advantage of quantum properties, could get a win rate rounding the 85% without resorting to hidden variables. Local realism proved to be not observable, contrary to quantum non-locality. Then, it was presented the quantum gates used in quantum circuits. At the end of the chapter, in order to show the utility of the content presented that far, we two examples that combine quantum properties and quantum computation were given: the quantum oracle and the Hardy State experiment, stressing this last one for being another experiment that contrasts the results obtained between EPR and quantum mechanics.

The following chapter is dedicated to quantum circuits and quantum algorithms. It starts by presenting the RSA algorithm, a very famous algorithm that consists in protecting message by the factorization of two prime numbers. Then, it is presented the three consequences that rule some basic principles governed by quantum algorithms: the no-cloning theorem, which states that it is impossible to create an exact copy of a qubit without destroying the original; superdense coding, a quantum computation

operation that allows the transmission of classical bits in a quantum channel using a single qubit; and quantum teleportation, a property that takes advantage of quantum entanglement to instantly teleport a qubit from one place to another (regardless the distance) using classical channels. The following section describes one of the most famous quantum cryptographical protocols: the BB84 protocol. This protocol is ruled by the use of random secret keys shared between two parties. These shared keys are originated by polarizing a sequence of bits given a certain basis. When the other party receives the polarized keys, it is performed a measurement given a certain basis. Then it is shared which bases were used by each party in order to check if they have chose the same basis. If they do, they keep the bit and if they do not, the second party scratches that bit out for further correction. In other words, if they have chose the same basis, it is guaranteed that they will get the correct message, but if that is not the case, they have to repeat the process in order to correct the basis, without sharing which initial bits were chosen. This proved to be a very safe manner to transmit information between parties since an eavesdropper could never spy the shared message without denouncing its presence due to the fact that if he did eavesdrop, he would disturb the qubits state from the first party and when both parties would start their checking process, the presence of a third party would be exposed due to the use of different bases. The next section is dedicated to quantum error correction: a procedure that corrects noisy data in quantum computers, namely, bit flip and phase flip errors. To correct the bit flip error, redundancy is employed so it is possible to identity which bit was flipped given a sequence of that same bit. Unfortunately, no-cloning theorem forbids this copy procedure. Nevertheless, it is possible to apply a new kind of measurement, a syndrome measurement, with the aid of *ancilla* bits in order to identify if an error has occurred or not. After noticing if an error has occurred, it is possible to correct it and get the intended state. The chapter ends with the explanation of one important algorithm in quantum computing history: the Simon's algorithm. The algorithm was designed to solve Simon's problem, which consists in finding the value of a variable by repeating a given function that has $n$ bits at the input and $n$ bits at the output, as few times as possible. Although classical approaches would take exponential time to solve the problem, depending on the number of bits used, quantum computers can take linear time, exposing the power and the advantage on using a quantum computer to solve certain types of problems.

The fourth chapter is dedicated to categorical quantum mechanics, a new formalism that came to simplify quantum circuits in diagram form and one of the main topics in this dissertation. It first became necessary to introduce category theory in order to understand the fusion between this theory and quantum mechanics. For that reason, the chapter starts by giving a historical background on the roots of category theory, i.e., how we went from the almost "untouchable" set theory, that could describe everything in mathematics using the concepts of sets containing objects and defining functions to category theory per se. This last one theory distinguishes itself from the previous one by being formed by objects and morphisms. Although it is not possible to look inside the objects in category theory, i.e., know what they really represent, we know how objects do relate to other objects. This way, category theory can overcome the troubles found in set theory like the Russel's paradox. It is also presented some fundamental properties in category theory, namely, the concepts of functors and natural transformations. Then, a smooth introduction in graphical language is presented accompanied by some linear properties and an

example of how represent the no-cloning theorem in a diagrammatically way. This introduction is done in order to shortly after merge quantum mechanics and category theory. At first, the marriage between something as weird or even intimidating as quantum mechanics with what mathematicians call abstract nonsense might not be the best idea. In fact, it is totally legitimate to doubt about this union. Nevertheless, it is when ZX-calculus jumps in that things start to get some sense. By being familiar with the translations between quantum circuits and ZX-diagrams, and taking advantage of some essential rules from the ZX-calculus, the diagrams become amazingly shorter and simpler to read and depict. To prove these advantages, it is presented an example of how a quantum oracle can be simply demonstrated and drawn in the form of categories.

## 5.2 Final Remarks

The advances made towards quantum mechanics and quantum computing in the last few years proved to be remarkable. From proving quantum non-locality to processing qubits and quantum algorithms on a quantum computer, quantum mechanics as much weird and spooky as it might be, proved to be a great asset for the future of Mankind. The creation of quantum computers that can run quantum algorithms that exponentially out-speed classical computers might be one of the biggest breakthroughs in history of computer science. Not only that, many more fields in our society have the potential to benefit from it like medicine, security, pharmaceutic and even in artificial intelligence fields where quantum neural networks and other machine learning programs are starting to emerge. And as any science field in existence, it comes a time to consider how much is possible to simplify the information we have. For big and complex quantum circuits comes innovative diagrams. Categorical quantum mechanics is a groundbreaking promising representation of quantum theory and for that reason, was purposed to be presented in this dissertation. Its advantages or even its understanding might not be so evident and clear as the authors of the book 'Picturing Quantum Processes: A first course in quantum theory and diagrammatic reasoning' claim to be. The main reason for this difficulty resides in the fact that Hilbert space-based language is quite present in our minds for which its translation to a new diagrammatic language is not that immediate. Furthermore, one cannot primarily see any difference between a quantum circuit and a categorical quantum diagram representing that same circuit. It is when the ZX-calculus comes into play that things start to get some sense and start to simplify quantum diagrams. The bottom line is that quantum picturalism can, indeed, capture essential features on quantum processes, eliminating some redundant data which might get over some obstacles found in equation-based languages. And that is a marvelous and very practical discovery. Moreover, this new language is starting to get attention of many researchers and stakeholders, even if it is necessary a strong basis on quantum mechanics and quantum information. However, only time will tell if this adaptation into a new representation language could lead quantum computing or even quantum mechanics into a brighter future.

It is somewhat bizarre how such a non-intuitive and abstract theory like quantum mechanics can be so much interesting, enigmatic and controversial. The more we learn about it, the more we do not understand; and the more we do not understand, the more we want to learn. It is intriguing to know

that there are so many secrets still to unveil about this theory and how those secrets will change the paradigm of quantum computing and computer science, which holds a particular interest in engineering.

As for future work it should be a good idea to stay up to date when it comes to quantum computation, specially if a discover is made with the aid of categorical quantum diagrams and ZX-calculus. With that being said, researches on this computer science field will be continued.

# Bibliography


[1] Junaid Rehman. *Advantages and disadvantages of quantum computers*. https://www.itrelease.com/2020/10/advantages-and-disadvantages-of-quantum-computers/ [Accessed: 16-11-2021].

[2] Wikiquote. *If you think you understand quantum mechanics, you don't understand quantum mechanics.* https://en.wikiquote.org/wiki/Talk:Richard_Feynman [Accessed: 15-05-2022].

[3] AZQuotes. *Quantum mechanics makes absolutely no sense.* https://www.azquotes.com/quote/905256 [Accessed: 15-05-2022].

[4] J. Clerk Maxwell. A dynamical theory of the electromagnetic field. pages 459–512, 1864. doi:https://royalsocietypublishing.org/doi/pdf/10.1098/rstl.1865.0008.

[5] The Nobel Prize. *The Nobel Prize in Physics 1921 - Albert Einstein*. https://www.nobelprize.org/prizes/physics/1921/summary/ [Accessed: 15-05-2022].

[6] The Nobel Prize. *The Nobel Prize in Physics 1927 - Arthur Holly Compton and Charles Thomson Rees Wilson*. https://www.nobelprize.org/prizes/physics/1927/summary/ [Accessed: 15-05-2022].

[7] David Mermin. *Boojums all the way through: communicating science in a prosaic age*. Cambridge University Press, 1990.

[8] Wikipedia The free encyclopedia. *Categorical Quantum Mechanics*, https://en.wikipedia.org/wiki/categorical_quantum_mechanics [accessed: 19-05-2022].

[9] The Nobel Prize. *The Nobel Prize in Physics 2022 - Alain Aspect, John F. Clauser and Anton Zeilinger*. https://www.nobelprize.org/prizes/physics/2022/press-release/ [Accessed: 21-10-2022].

[10] Pat Research. *What is quantum computing? Top 18 quantum computing companies*. https://www.predictiveanalyticstoday.com/what-is-quantum-computing/ [Accessed: 17-11-2021].

[11] Bob Coecke and Aleks Kissinger. *Picturing quantum processes*. Springer, 2017.

[12] Jagdish Mehra and Helmut Rechenberg. *The Quantum Theory of Planck, Einstein, Bohr and Sommerfeld: Its Foundation and the Rise of Its Difficulties 1900–1925 (The Historical Development of Quantum Theory, 1 / 2)*. Springer-Verlag, 1982.

[13] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982. doi:https://doi.org/10.1007/BF02650179.

[14] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. 400:97–117, 1985. doi:https://doi.org/10.1098/rspa.1985.0070.

[15] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, USA, 10th edition, 2011.

[16] Stephen Barnett. *Quantum Information*. Oxford University Press, Inc., USA, 2009.

[17] Eleanor G Rieffel and Wolfgang H Polak. *Quantum computing: A gentle introduction*. MIT Press, 2011.

[18] Ivan Djordjevic. Chapter 1 - introduction. In *Quantum Information Processing and Quantum Error Correction*. Academic Press, Oxford, 2012.

[19] Wikipedia The free encyclopedia. *Quantum logic gate*. https://en.wikipedia.org/wiki/Quantum_logic_gate [Accessed: 27-11-2021].

[20] Robert S Sutor. *Dancing with Qubits: How quantum computing works and how it can change the world*. Packt Publishing Ltd, 2019.

[21] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, May 1935. doi:10.1103/PhysRev.47.777.

[22] John. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1:195–200, Nov 1964. doi:10.1103/PhysicsPhysiqueFizika.1.195.

[23] Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24, Mar 1994. doi:https://doi.org/10.1007/BF02058098.

[24] John F. Clauser, Michael A. Horne, Abner Shimony, and Richard A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23:880–884, Oct 1969. doi:10.1103/PhysRevLett.23.880.

[25] David Mermin. *Quantum computer science: an introduction*. Cambridge University Press, 2007.

[26] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to Mathematical Cryptography*, volume 1. Springer, 2008.

[27] Erwin Schrödinger. The present status of quantum mechanics. *Die Naturwissenschaften*, 23(48):1–26, 1935. https://doi.org/10.48550/arXiv.2104.09945.

[28] Noson S Yanofsky and Mirco A Mannucci. *Quantum computing for computer scientists*. Cambridge University Press, 2008.

[29] QuTech Academy. *Quantum Key Distribution using BB84*, https://www.qutube.nl/quantum-internet/quantum-key-distribution-46 [accessed: 03-06-2022].

[30] Medium. *Quantum Key Distribution and BB84 Protocol*, https://medium.com/quantum-untangled/quantum-key-distribution-and-bb84-protocol-6f03cc6263c5 [accessed: 07-06-2022].

[31] Artur Ekert. *Introduction to Quantum Information Science Lecture*, https://www.youtube.com/c/arturekert/featured [accessed: 12-07-2022].

[32] Wikipedia The free encyclopedia. *Quantum Error Correction*, https://en.wikipedia.org/wiki/quantum_error_correction [accessed: 11-07-2022].

[33] Andrew M. Steane. A tutorial on quantum error correction. Centre for Quantum Computation, Department of Physics, University of Oxford, https://www2.physics.ox.ac.uk/sites/default/files/ErrorCorrectionSteane06.pdf.

[34] Ryan LaRose. Quic seminar . simon's algorithm. pages 49–54. https://www.ryanlarose.com/uploads/1/1/5/8/115879647/simon.pdf.

[35] David Deutsch and Richard Jozsa. Rapid solutions of problems by quantum computation. Royal Society of London, 1992. doi:10.1098/rspa.1992.0167.

[36] Wikipedia The free encyclopedia. *Nicolas Bourbaki*, https://en.wikipedia.org/wiki/nicolas_bourbaki [accessed: 09-03-2022].

[37] Wikipedia The free encyclopedia. *Set theory*, https://en.wikipedia.org/wiki/set_theory[accessed: 17-03-2022].

[38] Wikipedia The free encyclopedia. *Category theory*, https://en.wikipedia.org/wiki/category_theory[accessed: 18-03-2022].

[39] Chris Heunen and Jamie Vicary. *Categories for Quantum Theory: an introduction*. Oxford University Press, 2019.

[40] John van de Wetering. Zx-calculus for the working quantum computer scientist. December 2020. doi: https://doi.org/10.48550/arXiv.2012.13966.

[41] Wikipedia The free encyclopedia. *ZX Calculus*, https://en.wikipedia.org/wiki/zx-calculus [accessed: 02-09-2022].

[42] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *Society for Industrial and Applied Mathematics*, 26(5):1484–1509, 1997. doi:10.1137/S0097539795293172.

[43] G. Casati and G. Benenti. Quantum computation and chaos. In *Encyclopedia of Condensed Matter Physics Ch.9*, pages 9–17. Elsevier, Oxford, 2005.

[44] Quantum Information Portal and Wiki. *Shor's factoring algorithm*. https://www.quantiki.org/wiki/shors-factoring-algorithm/ [Accessed: 16-11-2021].

[45] Wikipedia The free encyclopedia. *Birthday Problem*, https://en.wikipedia.org/wiki/birthday_problem [accessed: 11-10-2022].

# Appendix A

# Quantum Mechanics Basic Notions

## A.1 The Bra-Ket Notation

The quantum gate NOT (Pauli Gate X) (2.16) can be written using the bra-ket notation as follows

$$X = |0\rangle \langle 1| + |1\rangle \langle 0| \tag{A.1}$$

This gate applied to a $|0\rangle$ must result in $|1\rangle$ according to its definition.

$$X|0\rangle = (|0\rangle \langle 1| + |1\rangle \langle 0|)|0\rangle = |0\rangle \langle 1|0\rangle + |1\rangle \langle 0|0\rangle \tag{A.2}$$

Since $\langle 1|0\rangle$ are orthogonal and $\langle 0|0\rangle$ are orthonormal

$$X = |0\rangle \cdot 0 + |1\rangle \cdot 1 = |1\rangle \tag{A.3}$$

## A.2 Projection Operator

Aside the Hermitian and the unitary operator, there is also the projection operator. A projection operator $P$ is given by

$$P = |\psi\rangle \langle \psi| \tag{A.4}$$

One should note that

$$P^2 = |\psi\rangle \langle \psi|\psi\rangle \langle \psi| = P \tag{A.5}$$

which means that projection $P$ is an idempotent operator.

Note that if we make

$$P|\psi\rangle = \lambda |\psi\rangle \tag{A.6}$$

and since $P$ is idempotent

$$P^2|\psi\rangle = \lambda^2 P|\psi\rangle = P|\psi\rangle = \lambda|\psi\rangle \tag{A.7}$$

From this result, we can conclude that

$$\lambda^2 = \lambda \tag{A.8}$$

which means $\lambda$ is either $0$ or $1$.

# Appendix B

# Demonstrations for Hardy's State

## B.1 Equivalency of expressions for the Hardy's state

Expressions (2.74) and (2.75) on Chapter 2 are equivalent. This equivalency can be demonstrated as follows:

$$
\begin{aligned}
|\Phi\rangle = \frac{1}{\sqrt{3}}\left(2|00\rangle - H_A H_B|11\rangle\right) &= \frac{1}{\sqrt{3}}\left(2|00\rangle - (H_A|1\rangle) \otimes (H_B|1\rangle)\right) \\
&= \frac{1}{\sqrt{3}}\left(2|00\rangle - \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)\right) \\
&= \frac{1}{\sqrt{3}}\left(2|00\rangle - \frac{1}{2}\left(|00\rangle - |01\rangle - |10\rangle + |11\rangle\right)\right) \\
&= \frac{1}{\sqrt{3}} \times \frac{1}{2}\left(4|00\rangle - |00\rangle + |01\rangle + |10\rangle - |11\rangle\right) \\
&= \frac{1}{\sqrt{12}}(3|00\rangle + |01\rangle + |10\rangle - |11\rangle
\end{aligned}
\tag{B.1}
$$

## B.2 Application of a Hadamard Gate for a Two Qubit State

A Hadamard gate applied to the state $|00\rangle$ is given by

$$
\begin{aligned}
(I \otimes H)|00\rangle &= \left[ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \right] [|0\rangle \otimes |0\rangle] \\
&= \begin{pmatrix} 1 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & 0 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ 0 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} & -1 \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\
&= \frac{1}{\sqrt{2}} \left[ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \right] \\
&= \frac{1}{\sqrt{2}} (|00\rangle + |01\rangle)
\end{aligned}
\tag{B.2}
$$

# Appendix C

# Shor's Algorithm

RSA algorithm encrypts messages by using a large number in such a way that decrypting it would require knowing the prime factors of that number. A classic computer could take years and a huge amount of computing resources to find those factors since the best method to do it is by guessing by trial and error. This does not make the RSA encryption invincible, it just shows that it is a really hard problem to solve. Nevertheless, in 1994 an American mathematician named Peter Shor (1959-) formulated an algorithm that could find those prime numbers in polynomial-time using a quantum computer. This algorithm that could break the RSA encryption was then published in 1997 and is known as the Shor's algorithm [42][43][25, Chapter 3]. Given a periodic function, Shor's algorithm calculates its period using a quantum Fourier transform. This transform is the key ingredient for quantum factoring. But what exactly is a quantum Fourier transform?

## C.1 Quantum Fourier Transform

One of the most useful ways to solve a mathematical problem is to transform it into some other problem with a known solution. The quantum Fourier transform enabled the construction of fast algorithms on quantum computers due to their fast computation. To study this transform, it is important to have a brief idea of roots of unity [20, Chapter 10]. A $N$-th root of unity, where $N$ is a positive integer, is a number $z$ that satisfies (C.1)

$$z^N - 1 = 0 \tag{C.1}$$

where each root is a number $\omega$ given by

$$\omega = \cos\left(\frac{2k\pi}{N}\right) + i\sin\left(\frac{2k\pi}{N}\right) = e^{\left(i\frac{2k\pi}{N}\right)} \tag{C.2}$$

These roots are called roots of unity because if each root is powered to the number of $N$, the result will be $1$. One should note that the sum of these roots will always be $0$ independently of the number $N$ and the product of the roots will be $1$ if $N$ is odd and $-1$ if $N$ is even.

Once introduced the roots of unity, we can build a discrete Fourier transform matrix. If we have $n$

bits, $N$ will be given by $N = 2^n$ and the roots $\omega$ will be still given by (C.2). With these values, we can compute an $N \times N$ DFT[1] matrix

$$\mathrm{W} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \omega^3 & \cdots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \omega^6 & \cdots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \omega^9 & \cdots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \omega^{3(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix} \tag{C.3}$$

where $\frac{1}{\sqrt{N}}$ is a normalization factor. Note that there is a pattern along each row of each individual column. For a better understanding of this computation, it is provided two examples: the first when using one bit and the second when using two bits.

For one bit we have

$$N = 2^1 = 2 \tag{C.4}$$

$$\omega = e^{\left(i\frac{2k\pi}{2}\right)} = e^{(ik\pi)} = -1 \Rightarrow \mathrm{R}(2) = \{1, \omega\} \tag{C.5}$$

The matrix $W$ will therefore by given by

$$W = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 \\ 1 & \omega \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{C.6}$$

This result is a very special case because we obtained the Hadamard gate, where the first column, of course, corresponds the $|+\rangle$ and the second column the $|-\rangle$.

For two bits we have

$$N = 2^2 = 4 \tag{C.7}$$

$$\omega = e^{\left(i\frac{2k\pi}{2}\right)} = e^{(ik\pi)} = i \Rightarrow \mathrm{R}(4) = \left\{1, \omega, \omega^2, \omega^3\right\} \tag{C.8}$$

The matrix $W$ will therefore by given by

$$\mathrm{W} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{pmatrix} = \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & 1 & \omega^2 \\ 1 & \omega^3 & \omega^2 & \omega \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \tag{C.9}$$

Note that the values for the different powers of $\omega$ are obtained through rotations on the unit circle. We always start at $1$ and go to $i$, $-1$, $-i$ or $1$ again depending on the power of $\omega$.

[1] Acronym for Discrete Fourier Transform

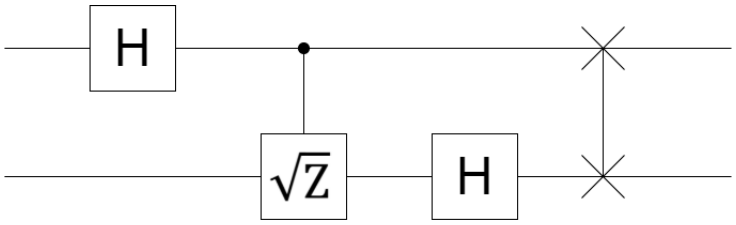



Figure C.1: Quantum circuit representation for a QFT[2] using two qubits

Now that we know how to calculate the DFT matrix $W$, we can apply the discrete Fourier transform which basically transforms a string of numbers $x_k$ in another string of numbers $X_k$.

$$X_k = \sum_{m=0}^{N-1} e^{\left(i\frac{2\pi}{N}mk\right)x_k} \Leftrightarrow X = Wx \tag{C.10}$$

Taking the previous example for $n = 2$ bits and for $x^T = (1, 2+i, i, -1-2i)$, the $X$ transform will be given by

$$X = Wx = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix}\begin{pmatrix} 1 \\ 2+i \\ i \\ -1-2i \end{pmatrix} = \begin{pmatrix} 1 \\ -1+i \\ i \\ 2-2i \end{pmatrix} \tag{C.11}$$

But what does the DFT has to do with the QFT[3]? Well, mathematically, the DFT and the QFT are exactly the same thing, i.e., the calculations used to compute the DFT matrix are exactly the same for the quantum Fourier transform. However, the secret of the QFT is its change of basis from a computational basis (e.g. $|0\rangle$,$|1\rangle$) to a Fourier basis (e.g. $|+\rangle$,$|-\rangle$) to speed up its computation. The relation between these two basis is given by (C.12).

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle) \tag{C.12}$$

As verified in (C.6), but this time with another notation, the gate that produces the quantum Fourier transform for one qubit is indeed the Hadamard gate.

$$\mathrm{U}_{\mathrm{QFT}}^{(1)} = \mathrm{H} \tag{C.13}$$

With some simple calculations we can easily see that

$$H|0\rangle = |+\rangle \qquad H|1\rangle = |-\rangle \qquad H|+\rangle = |0\rangle \qquad H|-\rangle = |1\rangle \tag{C.14}$$

Finally, we can easily understand that, by definition, the change of basis from a computational basis $|x\rangle$ to a Fourier basis $|\tilde{x}\rangle$ is given by

$$|\tilde{x}\rangle = \mathrm{U}_{\mathrm{QFT}}^{(n)}|x\rangle = \frac{1}{\sqrt{N}}\sum_{y=0}^{N-1}\exp\left(i\frac{2\pi xy}{N}\right)|y\rangle \tag{C.15}$$

[2]Acronym for Quantum Fourier Transform
[3]Acronym for Quantum Fourier Transform

where $n$ is, as always, the number of qubits and $U_{QFT}$ is a quantum unitary gate.

For the previous example $n = 1$ qubit ($N = 2$) and substituting it in expression (C.15) we have

$$|\tilde{x}\rangle = \mathrm{U}_{\mathrm{QFT}}^{(1)}|x\rangle = \frac{1}{\sqrt{2}}[|0\rangle + e^{i\pi x}|1\rangle] \tag{C.16}$$

Note that this new expression is affected by a phase, either when $x = 0$ and when $x = 1$.

$$x = 0\text{: } |\tilde{0}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle \qquad x = 1\text{: } |\tilde{1}\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle \tag{C.17}$$

This is another way to show that for one qubit, the quantum Fourier transform is the Hadamard gate. For two or more qubits, it is done similar calculations but the QFT is more complex to determine.

## C.2 Theoretical Explanation of Shor's Algorithm

Shor's algorithm is one of the most important algorithms in theory of computation. This algorithm made a huge difference in raising interest in quantum computation due to its speed by solving problems that classical computers could take years to find a solution. As stated before, Shor's algorithm can break the RSA encryption by calculating the period of a function using quantum superposition and quantum Fourier transform. Since it uses quantum mechanics, this algorithm is probabilistic, i.e., it gives the most probable answer which is most likely the correct one. While quantum superposition is used to simultaneously calculate an amount of possible solutions at once, the quantum Fourier transform is used to determine the period that will be used to find the factors of $N$.

Given a big number $N$ described as the product of two prime numbers, Shor's algorithm starts by guessing an arbitrary number that might share a factor with $N$ [44][25, Chapter 3]. Note that this guess does not need to be a pure factor of $N$, it could also be a number that shares factors with $N$ thanks to a simple but efficient method used to compute the greatest common divisor (GCD) of two integers called the Euclid's algorithm. Finding any share factor with Euclid's algorithm would mean that the factorization is solved because we could just divide $N$ by that factor to get the other factor and break the encryption. However, when it comes to big numbers, it is very unlikely that any guess will share a factor with $N$.

The initial guess will be then transformed into two more likely guesses that share factors with $N$. These guesses have origin in the mathematical fact described in (C.18)

$$a^c = m \cdot N + 1 \tag{C.18}$$

where $a$ and $N$ are integers that do not share factors, and $m$ is an integer. For example, if $a = 7$, $N = 15$ and $c = 2$

$$7^2 = 3 \cdot 15 + 4 \tag{C.19}$$

which does not match (C.18). It only matches if we power $a$ to $c = 4$

$$7^4 = 160 \cdot 15 + 1 \tag{C.20}$$

So, given a big number, $N$ and a initial guess $a$, it is mathematically guaranteed that expression (C.18) is verified. With some algebraic manipulation, we can write

$$a^c - 1 = m \cdot N \Rightarrow \left(a^{\frac{c}{2}} + 1\right) \cdot \left(a^{\frac{c}{2}} - 1\right) = m \cdot N \tag{C.21}$$

where $\left(a^{\frac{c}{2}} + 1\right)$ and $\left(a^{\frac{c}{2}} - 1\right)$ are the new and betters guesses that Shor's algorithm uses to find the factors of $N$. Of course, these guesses can still be multiples of factors of $N$ rather than $N$, but, if that is the case, we can find the shared factors by using Euclid's algorithm.

However, there are three essential problems with these guesses. The first problem is that one of the new guesses might itself be a multiple of $N$ and the other could be a factor of $m$ which would not be helpful for our goal. The second problem is that $c$ might be an odd number which will result in $\frac{c}{2}$ not being an integer number, making the guess $a$ not integer either. The third and more crucial problem resides in finding the value of $c$. If $N$ was small enough, a classic computer could solve the problem but realistically, $N$ is always a very big number.

On a quantum computer, it is possible to solve the decryption problem for a big number in an incredible speed by using quantum superposition to calculate simultaneously all possible solutions at once, finding the most likely answer and eliminating the others with destructively interference.

First, the quantum computer takes a number $x$ as an input, and raises $a$ to the power of $x$. Then it needs to take that result and calculate how much bigger than a multiple of $N$ it is. Let us call $r$ the difference between the factor of $N$ and the result. Note that we want $r$ to have the value $+1$ according to (C.18).

After that, it is made two kinds of superpositions: the first one is a superposition of all possible values of $x$ and the second one is a superposition of all possible values of $r$, where each $r$ results from its corresponded value $x$ of the first superposition.

As stated before, we now need to get all the incorrect guesses to destructively interfere and cancel out with each other's, leaving us with only one and the most likely answer $c$. This is a crucial point that makes the Shor's algorithm achievable only on a quantum computer. For that, we need to use another mathematical fact:

$$a^x = m \cdot N + r \quad \Rightarrow \quad a^{x+c} = m_2 \cdot N + r \tag{C.22}$$

Note that by adding something in the exponent of $a$, the only thing that changes on the right side of the equation is the value of $m$, keeping unchanged the values of $N$ and $r$. So, if we have the first superposition of all values of $x$ and just measure those who have for example $r = 3$ the result will be a superposition of all values of $x$ that have $r = 3$. The important part here is that all these values of $x$ repeat periodically with a period of $c$. In other words, they are "$c$" apart from each others.

If we find the frequency $\frac{1}{c}$, we can find $c$ and break the encryption and to find that frequency we will

be using the quantum Fourier transform. With this transform, all different frequencies will destructively interfere, leaving one single quantum state: the frequency $\frac{1}{c}$.

With the frequency $\frac{1}{c}$ we can easily calculate the period $c$, as long as $c$ is even, and apply it in (C.21). Even if we do not get an exact multiple of $N$, we can use Euclid's algorithm to find the factors and finally break the encryption.

In the end, Shor's algorithm sums up to this: for any arbitrary guess of a number that shares factors with $N$, that guess powered to $\frac{c}{2}$ and summing or subtracting $1$ is a much better guess if we know the value of $c$. While $c$ can be found using a classic computer (which would take a large amount of time) with a quantum computer it can be found almost immediately and that is why Shor's algorithm is one of the biggest breakthroughs in quantum computation.

# Appendix D

# Birthday Paradox

Imagine a scenario where 23 physicists are in a room. What is the probability of two of those physicists share a birthday? If you think that probability is low, your answer is incorrect. There is a $50\%$ chance of two of them have the same birthday [34] [45]. It seems absurd at the first impact but we can mathematically prove that it is in fact true (that is why Birthday paradox is consider a veridical paradox). Let us calculate the probability of two physicists not sharing a birthday (it is multiplied the individual probabilities of not sharing a birthday with any other physicist in the room).

$$\begin{aligned} P(\text{Not sharing a birthday}) &= \frac{365}{365} \times \frac{364}{365} \times \frac{363}{365} \times \cdots \times \frac{344}{365} \times \frac{343}{365} \\ &= \frac{364!}{342! \times 365^{23-1}} \approx 0.493 \end{aligned} \tag{D.1}$$

So, the probability of sharing a birthday is

$$P(\text{Sharing a birthday}) = 1 - 0.493 = 0.507 = 50\% \tag{D.2}$$

Note that this happens because each physicist is being evaluated with each other. Simon's algorithm works similarly since we too care about any pair. Each bit-string is being evaluated with another. If there are $n$ strings for available for choose, if one is randomly picked, it will remain $n - q$ strings. The total number of pairs of $m$ strings is

$$\frac{m(m-1)}{2} \approx m^2 \tag{D.3}$$

The probability of picking up two binary strings that share the same $f(x)$ is

$$\frac{1}{2^{n-1}} \approx \frac{1}{2^n} \tag{D.4}$$

So, the expected number of strings that share $f(x)$ is

$$\frac{m^2}{2^n} \tag{D.5}$$

which means that to have the chance of sharing $f(x)$ is

$$m = 2^{\frac{n}{2}} \tag{D.6}$$

hence, the reduction of the complexity of Simon's algorithm on a classical computer.