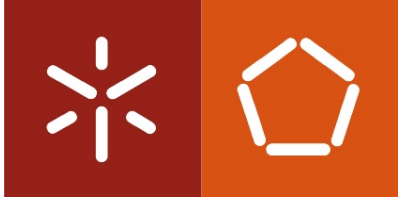
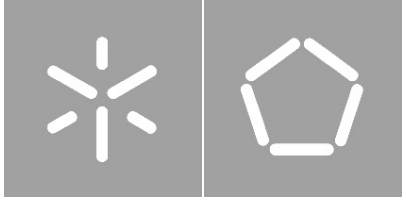


Universidade do Minho
Escola de Engenharia

Fernando Guilherme Gonçalves Pequeno de Oliveira e Silva

O impacto da computação quântica na Criptografia moderna

Janeiro de 2013



Universidade do Minho
Escola de Engenharia

Fernando Guilherme Gonçalves Pequeno de Oliveira e Silva

O impacto da computação quântica na Criptografia moderna

Tese de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de
Doutor José Carlos Bacelar Almeida
Doutor José Bernardo Barros

Janeiro de 2013

O impacto da computação quântica na Cripografia moderna

Fernando Guilherme Gonçalves Pequeno de Oliveira e Silva
(pg17942@alunos.uminho.pt)

Dissertação submetida na Universidade do Minho para obtenção do grau de Mestre em Engenharia Informática, elaborada sob a orientação de José Carlos Bacelar Almeida e José Bernando Barros.

Departamento de Informática
Escola de Engenharia
Universidade do Minho

Braga, 9 de Abril de 2013

Abstract

Quantum computing emerged from quantum physics as a new computational model. In a quantum computer, data is encoded directly in the physical state of a quantum system, and data-operations are transformations governed by the dynamics of quantum mechanics. The computational model obtained differ significantly from their classical counterpart, allowing to solve efficiently problems that are believed not to possess efficient solutions in classical computers. This fact has a deep effect in cryptography, since the security of most modern cryptographic schemes rely on hardness assumptions of particular problems, such as integer factorization or discrete logarithm.

This work aims to provide a comprehensive study of Quantum Computation and its mathematical foundation. Moreover, we will focus on the interaction between quantum computation and cryptography. Specifically, we will address: (1) the impact of some proposed quantum algorithms to the hardness assumptions of widely used cryptographic schemes, (2) new cryptographic "hard-problems"that are believed to be resilient to quantum computers, and (3) the exploitation of quantum effects in the design of new cryptographic schemes (quantum cryptography). The SAGE open source mathematics software system (www.sagemath.com) will be used to prototype/animate the concepts studied.

Resumo

A computação quântica emergiu da física quântica como um novo modelo computacional. Num computador quântico, a informação é codificada diretamente no estado físico de um sistema quântico, sendo as transformações deste, governadas pela dinâmica da física quântica. O modelo computacional obtido difere então de forma significativa do seu homólogo clássico, permitindo resolver de forma eficiente problemas, que se acredita não possuírem uma solução eficiente em computadores clássicos. Este facto produz um efeito profundo na criptografia, uma vez que a segurança da maior parte dos esquemas criptográficos modernos, baseia-se em assunções sobre a dificuldade de resolver determinados problemas no atual modelo computacional, como a fatorização de inteiros ou o logaritmo discreto.

Este trabalho procura então proporcionar um estudo abrangente sobre a computação quântica, bem como a sua fundamentação matemática. Além disso, vamos focar atenções na interação entre computação quântica e criptografia. Especificamente, vamos analisar (1) o impacto de alguns algoritmos quânticos, em assunções sobre a dificuldade de alguns esquemas criptográficos mais em uso atualmente, (2) novos esquemas criptográficos "problemas difíceis", que se acredita serem resistentes a computadores quânticos, e (3) explorar os efeitos quânticos no design de novos esquemas criptograficos - *criptografia quântica*. O sistema de software *open source* matemático SAGE (www.sagemath.com) será usado por forma a protótipar/animar alguns conceitos estudados.

Agradecimentos

Em primeiro lugar agradeço à 1ª arte. Sem o auxílio da música este trabalho teria sido penoso e claramente muito menos animado. Particularmente por isso, agradeço ao Mark Knopfler e aos acordes da sua guitarra, ao Chico Buarque e a bossa nova brasileira, ao Eric Clapton e os seus Blues e claro, ao quarteto de Liverpool(Beatles). As suas melodias transmitiram-me umas vezes a tenacidade necessária para não desistir e outras vezes a calma e o relaxamento para compreender.
Também quero agradecer à 7ª arte, porque no fim de um dia de trabalho (ás vezes até a meio) me permitia largar o problema em mãos e descansar, recarregando baterias para mais tarde voltar a investir. Aos meus amigos também, uma nota de agradecimento. Entre as venturas e desventuras que me proporcionaram, lá me foram incentivando a avançar, outras vezes a recuar, mas sem eles provavelmente este trabalho não seria igual.
Agradeço também à Sandra, amiga das matemáticas e ao meu orientador o professor Bacelar, por me terem ajudado a subir a montanha. Particularmente o professor Bacelar, companheiro de escalada, que nunca me deixou cair e me incentivou sempre.
Agradeço aos meus pais, que me fizeram do nada e constantemente me transmitem lições que preciosamente guardo (umas vezes mais resignado que outras). Agradeço à minha madrinha, o constante incentivo para me ver no mundo laboral ultrapassando para tal, este capítulo da minha vida. Por último, agradeço ao meu irmão e à Buddy (a cadela). O primeiro, por desdramatizar constantemente as várias dificuldades encontradas no percurso de escrever a dissertação. A segunda porque nunca me traiu e esteve sempre disponível para brincar e para me distrair.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Science offers the boldest metaphysics of the age. It is a thoroughly human construct, driven by the faith that if we dream, press to discover, explain, and dream again, thereby plunging repeatedly into news terrain, the world will somehow come clearer and we will gasp the true strangeness of the universe. And the strangeness will all prove to be connected, and make sense.

Edward O. Wilson

Nesta seção do documento, procuramos familiarizar ao leitor com uma visão global do que foi a construção deste trabalho. No decorrer deste processo, abordaremos questões pertinentes como a contextualização do tema de estudo a sua motivação e objetivos. Estas são as duas primeiras seções deste capitulo, cujo propósito é inerente ao seu significado. Além disto e por forma a proporcionar ao leitor uma visão global do trabalho, na seção sobre a estrutura do documento, discorreremos sobre os diferentes capítulos do mesmo. Convém ainda salientar, até pelo título desta tese de mestrado, que a mesma é um trabalho de investigação maioritariamente teórico. No entanto, procurou-se dotar o trabalho de uma escrita muito direcionada ao leitor, recorrendo frequentemente ao pronome pessoal ''nós", na tentativa de aproximar o raciocínio do leitor, àquele transmitido pelo autor na apresentação dos diversos conceitos.

1.1 Contextualização

A nossa compreensão do que nos rodeia esta fortemente ligada as leis da física, que acreditamos regerem o próprio universo. Esta busca pelo saber e compreensão do ambiente que nos rodeia é constantemente acicatada pela insaciável curiosidade humana. Com a descoberta do *Atmo* no século XIX, várias perguntas surgiram naturalmente sobre a constituição da matéria. No entanto, as leis da física não chegavam para compreender os fenômenos que à data se observavam. É então que no ínicio do século XX, surge na comunidade cientifica um novo ramo da física, que procurava explicar a dinâmica ou interação das partículas sub-atômicas - *mecânica quântica*. Este processo de descoberta (inicial) levou sensivelmente 20 anos, motivo pelo qual no fim de 1920 já existia uma modelo matemático concreto e aceite pela comunidade cientifica.
Durante meio século, os físicos procuraram aplicar este modelo no estudo das partículas e forças fundamentais. É deste esforço, que advém hoje os conhecimentos sobre *polímeros*, *semicondutores* e *supercondutores* entre outros. No entanto, e apesar destes desenvolvimentos nos permitirem avançar na nossa compreensão do que nos rodeia e no domínio tecnológico, pouco contribuíram para a compreensão da mecânica quântica em si.
No fim do século, concretamente nos anos 70 a perspectiva sobre a mecânica quântica alterou-se. Até então, ela era usada empiricamente isto é, como algo que se observava e depois se tentava explicar. Surgiu então a questão se não seria não podia ser algo que os experimentalistas conseguiriam recriar e controlar. É no seguimento destas questões, que se começa a ponderar se alguns dos problemas fundamentais das ciências da computação bem como, da teoria de informação, poderiam ser retratados na mecânica quântica. Nasce assim a computação quântica.
É no resto do século XX que a computação quântica ganha forma, com a apresentação de vários algoritmos que exploram as leis da mecânica quântica. Como veremos no decorrer deste trabalho, em alguns problemas específicos um algoritmo quântico consegue ter um desempenho extraordinário, em comparação com os seus homólogos clássicos conhecidos. Este desempenho é conseguido tanto em termos de capacidade efetiva de processamento, como de manipulação e armazenamento de informação. O entrave atual na sua aplicação direta na sociedade, prende-se com questões práticas, inerentes à construção de um computador quântico. Estas limitam a capacidade e tamanho dos computadores quânticos já existentes.

Uma outra área científica inerente a sociedade humana é a Criptografia. Como sabemos, o objetivo desta é tornar ilegível para terceiros, informação que se procura partilhar numa transmissão aberta, entre dois ou mais intervenientes. Ela é usada desde a antiguidade, explorando as mais variadas técnicas para esconder informação. A sua utilização nesses tempos idos, ia

desde a guerra até a ocultação de cartas amorosas. No entanto à medida que os tempos foram evoluindo, a noção de obscuridade começou a mudar. Resumidamente, nos primórdios da criptografia, procurava-se esconder o "design" das técnicas criptográficas, sendo este o fator de segurança do esquema. Isto revelou-se desastroso, em variados exemplos[1] ao longo história mundial. Mais recentemente, o paradigma de segurança mudou. Atualmente, o esquema criptográfico é publico, advindo a segurança da técnica da chave secreta que se usa.
Com o aparecimento dos computadores e de técnicas de criptoanálise[2] cada vez mais sofisticadas, as chaves secretas cresceram em tamanho bem como, em complexidade. Isto permitia proteger o material cifrado de ataques por força bruta - onde se percorre todo o universo de chaves. Além disso, as funções de encriptação e desencriptação passaram a basear-se em problemas que se acreditam serem difíceis de resolver computacionalmente, sem o conhecimento de todas as variáveis do problema. Como exemplo, temos o problema da fatorização de números inteiros ou o problema do logaritmo discreto.
Atualmente a criptografia encontra-se presente em todos os setores da sociedade moderna. Entre eles contam-se o governo, exército, banca e finanças. Este trabalho enquadra-se entre estas duas grandes aéreas e que estão em constante evolução e até interligadas. Com ele, procura-mos analisar o impacto que este novo modelo computacional poderá ter nas assunções criptográficas idealizadas para o modelo computacional clássico.

1.2 Objectivos

Devido a esta área estar muito ligada a física e a matemática, a compreensão da mesma por leitores curiosos de outros ramos das ciências, pode ser algo intimidatório. O presente trabalho teve então como objetivo delineado à partida, o estudo da matemática que sustenta a mecânica quântica, por forma a transmití-lo a esses leitores curiosos, nomeadamente do ramo das Engenharias. A pensar nas características do leitor, optou-se por uma exposição de conceitos mais prática (na medida do possível), apresentado sempre que possível exemplos simplistas dos mesmos.

Percebidos os conceitos da mecânica quântica, o objetivo seguinte passou por compreender que operações ou computações eram possíveis realizar num computador quântico. Poderiam recriar as mesmas operações lógicas que os seu homólogos clássicos, substituí-los? Ofereceriam alguma vantagem, em relação ao que conhecíamos das capacidades lógicas dos computadores clássicos?

[1] Máquinas Enigma, Segunda Guerra Mundial.

[2] Ramo da criptografia que se preocupa em descodificar informação sem que se tenha o conhecimento prévio da chave que a gerou.

De seguida traçamos rota para o algoritmo de *Shor*. Este interesse, deve-se à enorme implicação que o mesmo produziu nos esquemas criptográficos baseados no problema do logaritmo discreto e fatorização de inteiros. Estes esquemas, como por exemplo o *RSA*, são os mais usados a nível mundial. Por forma a atingir este objetivo, fez-se um levantamento das características do mesmo, percebendo as suas dependências e como estas eram colmatadas com recurso a diferentes conceitos e algoritmos quânticos. Nesse sentido apresentam-se esses mesmos algoritmos, procedendo a sua exposição com casos práticos sempre que possível.

Procurou-se ainda perceber qual a resposta que a comunidade criptográfica apresenta ao possível surgimento de um computador quântico. Neste percurso, concentramos atenções no uso de algoritmos quânticos, mas que desta vez produzissem protocolos de segurança como o protocolo *BB84* e *BB92*. Estes oferecem garantias de segurança muito fortes e que possivelmente solucionam a maior parte dos problemas para os quais pretendem ser solução. Existem na atualidade soluções comerciais dos mesmos. Para além disto, também apresentamos soluções baseadas em algoritmos clássicos já existentes e que se julgam ser imunes a computadores quânticos.

1.3 Estrutura do Documento

Esta seção do capítulo procura transmitir ao leitor a estrutura do trabalho que se segue. A mesma foi pensada para ter um fio condutor entre os diferentes conceitos, facilitando a sua compreensão. Na *Introdução à Mecânica Quântica*, procura-se rever todos os conceitos que se entenderam necessários para a compreensão dos princípios e algoritmos quânticos futuramente analisados. Por uma questão de coerência, optou-se por o dividir em dois. Numa primeira fase, discorre-se sobre os mais variados conceitos de álgebra linear necessários e numa segunda fase, intitulada de *postulados da mecânica quântica*, apresentam-se os respectivos postulados. Estes prendem-se com a necessidade de *mapear* os conceitos puramente matemáticos, numa entidade e ambiente concretos.
De seguida avançamos para *Circuitos Quânticos*. Este Capítulo serve dois propósitos. O primeiro que procura transmitir em detalhe o funcionamento lógico por detrás deste novo modelo computacional, realizável através de circuitos quânticos. O segundo, explica que existe um conjunto muito reduzido de operadores lógicos capazes de oferecer universalidade, tanto do ponto de vista clássico, como quântico.
A seguir em *Algoritmos Quânticos*, analisa-se um conjunto de algoritmos específicos, necessários para a compreensão do algoritmo de *Shor*. Este algoritmo faz parte de uma classe de algoritmos que dependem das propriedades da transformada de *Fourier*. Por seu lado, a transformada de *Fourier* depende de um conceito conhecido por paralelismo quântico, de onde advém

o desempenho extraordinário desta classe de algoritmos. Todos estes algoritmos são revistos nessa Seção recorrendo a pequenos exemplos práticos, sempre que possível.
Dada a importância do algoritmo de *Shor* dedica-se um Capítulo. Far-se-â uma revisão sobre Teoria de Números que lhe serve de base e enquadra-se esse problema numa rotina da transformada de *Fourier*. Apresenta-se também o conhecido exemplo da fatorização do número 15, descrevendo cada passo do mesmo. O maior impacto deste algoritmo incide sobre uma família criptográfica específica, mais precisamente, *criptografia assimétrica* que será revista no Capítulo sequente. No Capítulo *Impacto da Computação Quântica na Criptografia Atual*, analisa-se o impacto dos algoritmos quânticos nas diferentes famílias criptográficas. Na criptografia simétrica, não daremos um exemplo concreto como na assimétrica, mas mencionaremos o algoritmo quântico em questão bem como, o seu impacto e aplicação. Ainda nesse Capítulo, ver-se-á quais as soluções clássicas existentes que se acredita até ao momento, serem resistentes a computadores quânticos. Abordaremos ainda uma nova classe de esquemas criptográficos baseados em algoritmos quânticos. Nestes, procura-se dar um exemplo conciso da utilização de um deles, o algoritmo de acordo de chaves BB84.

Capítulo 2

Introdução à mecânica quântica

The most incomprehensible thing about the world is that it is comprehensible.

Albert Einstein

2.1 Fundamentos de Álgebra Linear

O estudo do enquadramento/modelo de mecânica quântica passa invariavelmente por um domínio prévio de alguns conceitos de álgebra linear. Este capítulo da dissertação, procura cobrir essa *base* algébrica e operacional. Além disto, procura introduzir a notação própria desta enquadramento/modelo, ao invés da que seria esperada da álgebra linear. Na seção de *postulados*, veremos como os conceitos matemáticos inicialmente introduzidos, são aplicados/*mapeados* num domínio físico específico, o *qubit*. As próximas páginas seguem de perto a exposição adotada em Nielsen and Chuang [2000].

Na álgebra linear tudo gira a volta de *vetores* e *operadores lineares*. Um espaço vetorial é formado por um conjunto de vetores, o seu elemento base, e possui uma dada *característica – dimensão*, *n*.

$$vector = \begin{bmatrix} z \\ \vdots \\ z_n \end{bmatrix} \tag{2.1}$$

A *dimensão* do espaço pode ser interpretada como o número de direções independentes desse espaço, dadas por $z_1 \dots z_n$. Como exemplo consideraremos, o espaço vetorial *complexo* de dimensão 2, dado por $\mathbb{C}^2$, inclui todos os pares

possíveis de números complexos. Como exemplo de vectores $\in \mathbb{C}^2$ temos:

$$v_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; v_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; v_{null} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.2}$$

O espaço vetorial onde definiremos os problemas aqui apresentados é o *espaço vetorial complexo*, mas para o definirmos corretamente precisamos de introduzir mais *conceitos/operadores* de álgebra linear. As restantes páginas desta seção servirão esse propósito. É possível consultar um resumo das mesmas na figura 2.1.

Notation	Description
z^*	Conjugado do numero complexo z.
$\|\Psi\rangle$	Vetor. Conhecido por *ket*.
$\langle\Psi\|$	Vector dual do $\|\Psi\rangle$. Conhecido por *bra*.
$\langle\varphi\|\Psi\rangle$	Produto interno entre os vectores $\|\varphi\rangle$ e $\|\Psi\rangle$.
$\|\varphi\rangle \otimes \|\Psi\rangle$	Produto tensorial entre $\|\varphi\rangle$ e $\|\Psi\rangle$.
$\|\varphi\rangle\|\psi\rangle$	Abreviação do produto tensorial entre $\|\varphi\rangle$ e $\|\psi\rangle$.
A^*	Conjugado da matriz A.
A^T	Transposta da matriz A.
$A^\dagger$	*Adjunta* da matriz A, dada por: $A^\dagger = (A^T)^*$ $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$.
$\langle\varphi\|A\|\Psi\rangle$	Produto interno entre $\|\varphi\rangle$ e $A\|\Psi\rangle$.

Tabela 2.1: Tabela com a *notações/operações* mais relevantes do modelo quântico.

2.1.1 A Base

Como foi mencionado os elementos base de álgebra linear são vetores. Na mecânica quântica estes possuem uma notação própria, $|\rangle$ e $\langle|$, conhecida por notação de *Dirac*. Denominados respectivamente *ket* e *bra*. De seguida, e para o mesmo vetor, apresentam-se as duas notações.

$$|\Psi\rangle = \begin{bmatrix} \Psi_0 \\ \Psi_1 \\ \vdots \\ \Psi_n \end{bmatrix}; \qquad \langle\Psi| = [\Psi_0^*, \Psi_1^*, \cdots, \Psi_n^*];. \tag{2.3}$$

onde a operação *****, corresponde ao complexo conjugado de cada entrada do vector $|\rangle$. De uma forma mais abstrata podemos definir esta notação por,

$$|v\rangle = \sum_i a_i |v_i\rangle \tag{2.4}$$

$$\langle v| = \sum_i a_i^* \langle v_i|. \tag{2.5}$$

Num espaço vetorial existe sempre um conjunto particular de vetores que *geram*[1] esse espaço vetorial. Isto é, qualquer vetor nesse espaço vetorial pode ser definido como uma combinação linear dos vetores desse conjunto. Uma maneira intuitiva e acessível de encontrar este conjunto, para um dado espaço vetorial, é procurar os vectores unitários[2] do mesmo. Como exemplo e considerando o espaço vectorial complexo $\mathbb{C}^2$, temos:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; \tag{2.6}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \tag{2.7}$$

Podemos então neste momento escrever um qualquer vector $|v\rangle$ de coordenadas genéricas ,

$$|v\rangle = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \tag{2.8}$$

como combinação de $|0\rangle$ e $|1\rangle$, da seguinte forma:

$$|v\rangle = a_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \tag{2.9}$$

Em termos gerais, uma *Base* é um conjunto de vetores *linearmente independentes*[3], que *geram* um determinado espaço vetorial. Uma forma de verificar rapidamente se um conjunto de vetores é *linearmente independente* é construir uma matriz, fazendo de cada vector, uma coluna da matriz. Reconsiderando o exemplo passado obtemos:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.10}$$

e de seguida, *verificamos se o determinante da matriz não é nulo*, o que é verdade para $\mathbf{A}$.[4] Posto isto, podemos aceitar $|v_1\rangle$ e $|v_2\rangle$ como uma *base* para $\mathbb{C}^2$. Mais ainda, o número de elementos da *base*, coincide com dimensão do espaço vetorial que esta define. Isto é facilmente observado no exemplo transato.

[1] do ingês *span*

[2] vectores de comprimento 1.

[3] Dado um conjunto de vectores é impossivel escrever cada um deles como uma combinação linear dos outros.

[4] DET(A) = 1.

2.1.2 Operadores lineares

Quando começamos por falar em espaços vetoriais, dissemos que os dois elementos basilares deste eram, *vetores* e *operadores lineares*. Após termos apresentado os conceitos importantes sobre os vectores bem como, a notação a usar para estes, vamos agora analisar os *operadores lineares* seguindo a mesma lógica.
De uma maneira geral, podemos ver um operador linear como uma entidade/função capaz de manipular vectores. Rigorosamente podemos defini-lo da seguinte forma:

$$A : V \to W \tag{2.11}$$

$$A\left(\sum_i a_i|v_i\rangle\right) = \sum_i a_i A(|v_i\rangle) \tag{2.12}$$

Da álgebra linear sabemos que os mesmos possuem uma representação matricial isto é, o comportamento de um operador linear pode ser codificado numa matriz.

$$\begin{bmatrix} a_{11} & a_{12} & ... \\ a_{21} & a_{22} & ... \\ ... & ... & ... \end{bmatrix}_{m\times n} \tag{2.13}$$

Sabe-se também, que a aplicação de um operador a um vector é vista como a multiplicação do operador (matriz) pelo vector. Considerando o operador **A** bem como, a definição formal de um operador linear, verificamos para um qualquer vector $\mathbb{V} \in \mathbb{C}^n$, que o dito operador o transforma num elemento/vetor $\mathbb{W} \in \mathbb{C}^m$. Esta propriedade é herdada pela definição do próprio operador. Da nossa parte, na construção de um operador linear temos apenas de nos preocupar com têS coisas:

- Comportamento do Operador,
- Base de Entrada,
- Base de Saida.

Assumindo como *Base* a apresentada da Seção anterior $|0\rangle$ e $|1\rangle$. Temos apenas de nos preocupar em definir em que medida (comportamento) queremos que cada operador manipule os vectores. Vamos consolidar estes conceitos com um exemplo extremamente simples e intuitivo. Retomando os vectores $|0\rangle$ e $|1\rangle$ pertencentes ao espaço vectorial $\mathbb{C}^2$, mais ainda estes vectores como sabemos, formam uma base desse espaço vectorial. O operador *Identidade* quando aplicado nesta base, deve refletir o seguinte comportamento.

$$I|0\rangle = |0\rangle \tag{2.14}$$

$$I|1\rangle = |1\rangle \tag{2.15}$$

Vamos então definir o comportamento do dito operador. Pela ação esperada do operador **I**, atendendo ao fato que v_1 e $v_2 \in \mathbb{C}^2$, verificamos que o operador **I** tem que ser a matriz identidade de dimensão 2×2, uma vez que o espaço vectorial de *saida* é igual ao de *entrada*. Assim sendo, **I** é dado por:

$$I_{2\times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.16}$$

e a sua ação é facilmente comprovada na Base escolhida:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \Rightarrow |0\rangle \tag{2.17}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow |1\rangle \tag{2.18}$$

O mesmo se verifica para qualquer vector, definido nesta base. Seja,

$$|\psi\rangle = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} a \\ b \end{bmatrix} \tag{2.19}$$

aplicando o operador $I_{2\times 2}$ obtemos:

$$|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} a \\ b \end{bmatrix} \equiv \begin{bmatrix} a \\ b \end{bmatrix} \equiv |\psi\rangle. \tag{2.20}$$

Apesar do comportamento trivial, este operador permite-nos ter alguma intuição de como na prática vetores e operadores interagem. Falta agora enunciar, quais as regras que os mesmo têm de obedecer de maneira a encaixarem no modelo de computação quântica. Por incrível que pareça, apenas têm de implementar uma regra. Serem *Unitarios*! Ou seja:

$$U^\dagger U = U U^\dagger = I \tag{2.21}$$

onde, $U^\dagger = (U^*)^T$ é conhecido como a operação *adjunta* ou *conjugada Hermitiana* sendo *, o símbolo matemático para o cálculo do *complexo conjugado* e "T" a operação de transposta. A aplicação desta operação é facilmente compreendida com um exemplo:

$$\begin{bmatrix} 1+3i & 2i \\ 1+i & 1-4i \end{bmatrix}^\dagger = \begin{bmatrix} 1-3i & 1-i \\ -2i & 1+4i \end{bmatrix}. \tag{2.22}$$

É fácil de comprovar que $I_{2\times 2}$ é um operador unitário. Vamos agora apresentar outro operador, com a finalidade de apresentar um método de construção do *output* pretendido, muito simplista. Pretende-se que o operador X, seja capaz de trocar os elementos da *base*. Ou seja,

$$X|0\rangle = |1\rangle \tag{2.23}$$

$$X|1\rangle = |0\rangle \tag{2.24}$$

isto vai provocar que qualquer vector definido à custa desta *Base* tenha as suas componentes trocadas isto é,

$$X|\psi\rangle \equiv X \begin{bmatrix} a \\ b \end{bmatrix} \equiv \begin{bmatrix} b \\ a \end{bmatrix} \tag{2.25}$$

Tomando como ponto de partida o que dissemos sobre o operador **A**, ou seja que este era responsável por enviar vectores de $\mathbb{C}^n$ para o espaço $\mathbb{C}^m$, percebemos que cada coluna da matriz desempenha uma ação na *base*. Nomeadamente, se observamos em maior detalhe verificamos que, cada coluna do operador é na realidade um elemento da *base* ou melhor, descreve a ação deste, num elemento particular da *base*. Vamos considerar a primeira coluna responsável pelo elemento da base $|0\rangle$ e a segunda por $|1\rangle$. Pretendemos que quando a entrada for $|0\rangle$ o operador transforme este vetor em $|1\rangle$ e vice versa. Como cada coluna descreve explicitamente o comportamento em cada elemento da base, acabamos de construir o operador X!.

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{2.26}$$

Ao analisarmos a ação do operador de acordo com o inicialmente pretendido, verificamos que este se comporta como esperado. Facilmente comprovamos também que o operador é unitário, visto que respeita a Equação 2.21. No curso deste trabalho, serão apresentados mais operadores, inclusive para diversas bases computacionais, no entanto esta abordagem de seguir pelas colunas de cada operador a transformação que queremos aplicar, mantêm-se.

2.1.3 Produto interno

Nesta seção, vamos finalmente definir completamente o espaço vectorial onde definiremos os nossos problemas quânticos. Mas antes disso, precisamos de compreender mais um conceito de álgebra linear nomeadamente, o *produto interno*. Esta operação, é conhecida por *mapear* dois vetores, num número complexo (no nosso caso). Esta operação introduz três importantes conceitos no espaço vectorial:

1. O comprimento de um vector.
2. O ângulo entre dois vectores.
3. A *norma* de um vector.

Sendo esta última, a responsável por transformar o *espaço vectorial complexo* de dimensão n - $\mathbb{C}^n$, num espaço de *Hilbert*, onde se definem os problemas do modelo de mecânica quântica. No nosso caso, interessa-nos a versão finita de espaços de *Hilbert*.

Voltando a operação do produto interno, podemos definí-la da seguinte forma:

$$\langle\psi|\phi\rangle :: V \times V \Rightarrow C \tag{2.27}$$

$$\langle\psi|\phi\rangle = \sum_i y_i^* z_i = \begin{bmatrix} y_1^* & \cdots & y_n^* \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}, t.q. y_i \in \langle\psi|, z_i \in |\phi\rangle \tag{2.28}$$

$$\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}, \tag{2.29}$$

que como foi dito retorna um número complexo. Mas qual é a nossa sensibilidade para com este valor, o que é que nos diz? Talvez a mais importante utilização desta operação, seja a de descobrir se dois vectores são ortogonais. Esse resultado traduz-se por,

$$\langle v_1|v_2\rangle = 0. \tag{2.30}$$

Na Seção de postulados veremos como este simples resultado é de suma importância.

2.1.4 Produto externo

Através da operação do produto interno apresentada na seção anterior, vamos agora introduzir uma notação muito útil, para representar um operador linear. Esta representação tem por nome *produto externo.*. Considere-se dois vetores, respectivamente, $|v\rangle$ no espaço de *Hilbert* $\mathbb{V}$ bem como, o vetor $|w\rangle$ que se encontra no espaço de *Hilbert* $\mathbb{W}$. Definimos o operador $|w\rangle\langle v|$ como:[5]

$$(|w\rangle\langle v|)(|v'\rangle) \equiv |w\rangle \langle v|v'\rangle = \langle v|v'\rangle |w\rangle. \tag{2.31}$$

Ou seja, o operador definido por $|w\rangle\langle v|$, 'mapeia" um vetor do espaço vetorial V no espaço W. Concretamente, esta notação permite-nos exprimir o resultado do operador $|w\rangle\langle v|$ quando este atua no vetor $|v'\rangle$. Analisando a Equação 2.31 percebemos que isto é equivalente a multiplicar o vetor $|w\rangle$ pelo numero complexo resultante da operação do produto interno $\langle v|v'\rangle$. Mais ainda, podemos exprimir esta operação através de combinações lineares,

$$\left(\sum_i a_i |w_i\rangle\langle v_i|\right)|v'\rangle \equiv \sum_i |w_i\rangle \langle v_i|v'\rangle . \tag{2.32}$$

Seja $|i\rangle$ uma qualquer base ortogonal para o espaço vectorial V. Podemos por isso escrever um vector $|v\rangle$ desse espaço vectorial, como uma combinação linear da mesma. Respectivamente, $|v\rangle = \sum_i v_i|i\rangle$ onde v_i, representa um determinado número complexo. Note-se que $\langle i|v\rangle = v_i$ e como tal,

$$\left(\sum_i |i\rangle\langle i|\right)|v\rangle = \sum_i |i\rangle \langle i|v\rangle = \sum_i v_i|i\rangle = |v\rangle. \tag{2.33}$$

[5] notação do produto externo

como $|v\rangle$ é genérico advém que para qualquer vector se verifica a igualdade anterior e portanto:

$$\sum_i |i\rangle\langle i| = I. \tag{2.34}$$

É através desta relação de *completude* que provém a aplicação desta representação para qualquer operador linear. Genericamente e considerando os espaços vetoriais descritos, definimos então o aplicação de um qualquer operador $\mathbb{A}$, para a sua representação através do produto externo,

$$\begin{aligned} A &:: V \to W \\ A &= I_W A I_V \\ &= \sum_{ij} |w_j\rangle\langle w_j|A|v_i\rangle\langle v_i| \\ &= \sum_{ij} \langle w_j|A|v_i\rangle |w_j\rangle\langle v_i|. \end{aligned} \tag{2.35}$$

Genericamente sobre $\mathbb{C}^2$, podemos observar esta aplicação como,

$$\begin{aligned} A &\equiv \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ &\equiv a_{11}|0\rangle\langle 0| + a_{12}|0\rangle\langle 1| + a_{21}|1\rangle\langle 0| + a_{22}|1\rangle\langle 1|. \end{aligned} \tag{2.36}$$

onde cada elemento do produto externo é calculado pelo somatório. Por exemplo, $a_{12}|0\rangle\langle 1|$ resulta de,

$$\langle 0|A|1\rangle|0\rangle\langle 1| \equiv \begin{bmatrix} 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{21} \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} |0\rangle\langle 1| \equiv a_{12}|0\rangle\langle 1|. \tag{2.37}$$

e de forma similar se obtêm os restantes termos. Como caso prático desta aplicação seguem-se dois exemplos. Considere-se o seguinte operador,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Então recorrendo a operação de *produto externo* podemos definir $\mathbb{X}$ por

$$X \equiv |0\rangle\langle 1| + |1\rangle\langle 0|. \tag{2.38}$$

Por outro lado, podemos exprimir o operador I da seguinte forma,

$$I \equiv |0\rangle\langle 0| + |1\rangle\langle 1|. \tag{2.39}$$

2.1.5 Valores e vectores próprios, observáveis, projectores

Quando um operador linear atua num vetor, normalmente altera-o na sua direção e magnitude. Contudo, quando um operador linear apenas altera a magnitude de um dado vetor, dizemos que esse vetor é um *vetor próprio* ou *Eigenvector*, desse operador linear. Esse mesmo vetor é paralelo ao original (direção mantém-se), diferindo do inicial por um escalar designado por *Eigenvalue* ou valor próprio que é definido pela seguinte equação:

$$A|v\rangle = \lambda|v\rangle, \lambda \text{ é um número complexo conhecido por valor próprio.} \quad (2.40)$$

Uma vez que os vetores próprios estão relacionados com os operadores, deve haver forma de os calcular. Para um operador $\mathbb{A}$, esse cálculo traduz-se na resolução da seguinte equação, conhecida por *equação característica*:

$$C(\lambda) = det|A - \lambda I| = 0 \quad (2.41)$$

onde **det**, corresponde a função que devolve o determinante de uma matriz. As soluções desta equação é um conjunto de λ_i, que são os diferentes valores próprios para os diferentes vetores próprios, que são mais tarde calculados. Toda esta operação, calculo dos valores próprios + vetores próprios é rapidamente obtida para um dado operador, recorrendo ao auxílio da plataforma *SAGE*. Dado um operador A, os métodos *A.eigenvalues()* e *A.eigenvectors_left()*, realizam isto mesmo. Particularmente, o método eigenvectors_left() devolve uma lista de triplos onde em cada triplo, o primeiro elemento é o valor próprio do operador, o segundo elemento é o vector próprio e o terceiro a multiplicidade.

Esta sintáxe, permite-nos introduzir mais um conceito, o de um operador *Observável*. Quando os *eigenvalues* de um determinado operador são números reais, dizemos que estamos na presença de um operador *Observável*. Esta propriedade, está relacionada com o facto de o espaço vetorial *Hilbert*, por nós usado, possuir dimensão finita. Intuitivamente, *Observável* implica algo que pode ser visto. Imaginemos que no nosso espaço vectorial possuímos vetores que codificam grandezas físicas, como as Forças (*gravidade*, *aceleração*,...). Então podemos imaginar um Observável como um operador que nos permite extrair uma medida ou quantidade sobre um vetor. Este possui as seguintes propriedades:

1. Se no espaço vetorial **V**, os vetores $|v_1\rangle$, $|v_2\rangle, \ldots |v_n\rangle$, codificarem uma determinada grandeza física, então pela equação 2.40 sabemos que os valores obtidos pelo observável **O**, $\lambda_1, \lambda_2, \ldots, \lambda_n$, serão valores reais. Tal está de acordo com a intuição referida em cima, de quantidades que podem ser medidas.

2. Da aplicação do operador **O** ao vetor $|v_1\rangle$, pela equação 2.40, apenas vamos obter o *eigenvalue* λ_1. Se a isto juntarmos o facto do operador

atuar numa determinada *Base*, sabemos pela dita equação que o vetor de *entrada* difere do *saida* pelo dito escalar. Ou seja, preserva-se as propriedades dos elementos da *Base*. O conjunto de vetores próprios de um *observável* é ortogonal.

3. O conjunto $v_1 \dots v_n$ de vetores próprios de um observável **O**, forma então uma *Base*.

Podemos então exprimir mais rigorosamente um observável $\hat{O}$,[6]como:

$$\hat{O} = \sum_n \lambda_n |v_n\rangle. \tag{2.42}$$

Este tipo de operador é conhecido por ser hermitiano ou auto adjunto. Isto é,

$$O^\dagger = O. \tag{2.43}$$

Esta nova terminologia, permite-nos introduzir uma nova classe de operadores conhecidos por *Projectores*, $|\rangle\langle|$. O seu comportamento pode ser exemplificado intuitivamente como:

$$P = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x+y \\ 0 \end{bmatrix}. \tag{2.44}$$

Neste exemplo, vemos um *projector* **P**, que atua num espaço vectorial de duas dimensões, *projectar* a componente **y** na componente **x** do vector. Mais formalmente, vamos considerar que temos dois espaços vectoriais **V** e **W** com dimensão **d** e **k** respectivamente, tal que, W $\subseteq$ V. Recorrendo a um método matemático chamado *Gram-Schmidt*(Anexo B.1) é possível construir uma base ortogonal $|1\rangle, \dots, |d\rangle$ para V tal que, $|1\rangle, \dots, |k\rangle$ forma uma base ortogonal para W. É então possível construir e definir um projetor do espaço vectorial V para o sub-espaço vectorial W, dado por:

$$P = \sum_{i=1}^{k} |i\rangle\langle i|. \tag{2.45}$$

Por último, um operador diz-se *normal* se,

$$AA^\dagger = A^\dagger A. \tag{2.46}$$

Facilmente comprovamos que qualquer operador *Hermitiano* é normal. Este novo conceito permite-nos introduzir um teorema importante,

Teorema 1. ***(Decomposição Espectral)****: Qualquer operador M normal definido num espaço vectorial V é diagonal relativamente a uma qualquer base ortogonal V. Por outro lado, qualquer operador diagonízavel é normal.*

cuja prova se encontra em Nielsen and Chuang [2000].

[6]sintaxe correcta

2.1.6 Produto tensorial

A operação do produto *tensorial*[7] ($\otimes$) é uma operação que permite juntar espaços vetoriais, de maneira a formar um espaço vetorial de maior dimensão. Ou seja, considere-se $\mathbb{V}$ e $\mathbb{W}$ espaços vetoriais de dimensão n e m respetivamente. Vamos também assumir que estes espaços são *espaços de Hilbert*[8]. Então $\mathbb{V} \otimes \mathbb{W}$, que se lê $\mathbb{V}$ "tensor" $\mathbb{W}$, resulta num espaço vetorial de dimensão **mn**.
Para nós, esta operação vai ser usada no que toca a vetores e matrizes. Ela vai permitir estender as propriedades que já conhecemos, para espaços vetoriais de maiores dimensões. Como por exemplo a questão da ortogonalidade das bases entre dois espaços vetoriais. Ou seja, considerando novamente os espaços vetoriais em cima apresentados, vamos supor que em $\mathbb{V}$ existe a *base* $|1\rangle, \ldots, |n\rangle$ e em $\mathbb{W}$ a base $|1\rangle, \ldots, |m\rangle$. Então ao fazermos $(|1\rangle, \ldots, |n\rangle) \otimes (|1\rangle \ldots |m\rangle)$, obtemos uma base para $\mathbb{V} \otimes \mathbb{W}$.
Se generalizarmos um vetor como uma matriz com uma coluna, podemos reduzir o *Tensor* de matrizes e vetores ao chamado produto de *Kronecker* de uma matriz $\mathbb{A}_{n\times m}$ por uma outra $\mathbb{B}_{p\times q}$, tal que,

$$A \otimes B = \overbrace{\left\{ \begin{bmatrix} A_{11}B & A_{12}B & \ldots & A_{1n}B \\ A_{21}B & A_{22}B & \ldots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \ldots & A_{mn}B \end{bmatrix} \right\}}^{n\times q} m \times p \tag{2.47}$$

Na seção de *postulados* veremos a aplicação desta operação. No caso de usarmos operadores lineares a notação usada será

$$A \otimes B \tag{2.48}$$

caso estejamos a trabalhar com vectores,

$$|\varphi\rangle \otimes |\psi\rangle. \tag{2.49}$$

2.2 Postulados da mecânica quântica

Até ao momento, abordamos os mais importantes conceitos de álgebra linear que precisamos para o estudo da mecânica quântica. Existem certamente aspectos da mesma que escaparam a esta revisão, mas que serão introduzidos pontualmente quando necessário. Posto isto, vamos agora focar atenções num conjunto de postulados, cuja finalidade é fazer a ponte entre o modelo físico e o modelo de mecânica quântica. Estes postulados vão ser analisados tendo como caso prático – *o qubit*, ou *quantum bit*, homólogo do seu "parente" clássico. Vamos então aproveitar o momento para introduzi-lo:

[7] do ingês tensor product.

[8] Se nada for dito em contrário vamos assumir sempre que o espaço vetorial em que nos encontramos é de *Hilbert*.

2.2.1 Qubits

Utilizando como ponto de partida a frase em cima proferida, "*o Quantum bit é o homólogo do seu "parente" clássico*", vamos começar por definir *qubit*. Como sabemos um bit clássico pode estar em apenas um de dois estados, 0 ou 1. Fazendo a ponte com o mundo quântico bem como, com toda a seção introdutória desta dissertação, podemos definir:

$$0 \Rightarrow \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \tag{2.50}$$

$$1 \Rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle. \tag{2.51}$$

como a *base* computacional do modelo de computação quântica, à semelhança da sua homóloga clássica. No entanto, as semelhanças entre os dois modelos computacionais não vão muito mais longe. Relembrando o que foi dito na seção 2(Base), mais concretamente, que no espaço vetorial $\mathbb{C}^2$ os elementos/-vetores $|0\rangle$ e $|1\rangle$ formavam uma *base* que *cobria* ou *gerava* todo esse espaço vetorial. Podemos inferir que na realidade um *qubit*, para além dos dois estados apresentados possui um terceiro, conhecido por *sobreposição*:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.52}$$

onde, α e β são números complexos.
Para estar de acordo com o que foi dito atrás, nomeadamente ser unitário, este vector $|\phi\rangle$, deve estar sujeito a seguinte condição de normalização:

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.53}$$

Ou seja, um *qubit* pode estar nos estados base $|0\rangle$ ou $|1\rangle$, ou então num estado de *sobreposição*, da combinação linear em cima referida. Isto é incrível, a capacidade de armazenamento de um único *qubit* é de longe superior a de um bit. Veremos no decorrer dos postulados, que apesar de isto ser verdade, ela não nos está diretamente acessível.

2.2.2 Espaço de Estados

Considere-se a seguinte transcrição de Nielsen and Chuang [2000][P.80]:
"*Associated to any isolated physical system is a complex vector space with an inner product (that is, a Hilbert space) known as the state space of the system. The System is completely described by its state vector, which is a unit vector in the system's state space.*"

Este postulado, contextualiza o espaço onde os problemas por nós apresentados são definidos no modelo da mecânica quântica. Começando pelo início, afirmamos que a nossa partícula ou sistema físico é o *qubit*. Neste caso, o *qubit* é representado por um *vetor de estado* de dimensão 2. O espaço vectorial onde este é definido é como foi enunciado, um *o espaço de Hilbert*. Considerando $|0\rangle$ e $|1\rangle$ como a base ortogonal para esse espaço de estados, podemos definir o qubit como:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle. \tag{2.54}$$

onde, α e β são números complexos. Mais ainda, a condição que $|\psi\rangle$ seja um vector unitário traduz-se em,

$$\langle\psi|\psi\rangle = 1 \equiv \|\alpha\|^2 + \|\beta\|^2 = 1 \tag{2.55}$$

Para nós, o *qubit* será sempre uma entidade abstrata. Apesar de este possuir uma representação no mundo físico, tal escapa ao domínio deste trabalho. Além disso, o modelo de mecânica quântica não nos diz para um *qubit* ou para outra partícula física, qual é o espaço de vetorial correto para a enquadrar ou mais especificamente qual o vetor de estados da mesma. Descobrí-los, cabe ao experimentalista/físicos e é considerado um problema difícil.

2.2.3 Evolução

Considere-se a seguinte transcrição de Nielsen and Chuang [2000][P.81]:
"*The evolution of a closed quantum system is described by an unitary transformation, that is, the state $|\psi\rangle$ of the system at time t_1 is related to state $|\psi\rangle'$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2 .*"

Este postulado centra-se na evolução ou melhor, especifica como esta ocorre, para um qualquer estado quântico. Em primeiro lugar, afirma que a mesma ocorre segundo uma operação unitária, mais concretamente, um operador unitário como analisamos na seção transata. Contudo, não nos diz que operadores unitários podemos usar. No nosso caso -*qubits*, qualquer operador unitário é passível de se aplicar num sistema realista. Como exemplo disso, temos o operador $\mathbf{X}$, e cuja função real, na aplicação a um *qubit* é a mesma

do seu homólogo clássico *"not gate"*. Outro operador muito utilizado, é o **Hadamard**:

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.56}$$

Vamos analisar muito rapidamente este operador, de maneira a tentar compreender o fenômeno de sobreposição:

$$H|0\rangle = \frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \tag{2.57}$$

$$H|1\rangle = \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}. \tag{2.58}$$

Facilmente percebemos, que este operador tem a capacidade de colocar um estado base $|0\rangle$ ou $|1\rangle$, num estado diferente. Intuitivamente, podemos afirmar que o resultado do operador, deslocou o estado base para "algures" entre $|0\rangle$ e $|1\rangle$. Com este exemplo, conseguimos demonstrar o enunciado do postulado, visto que podemos *relacionar* um estado, $|1\rangle$, ou $|0\rangle$ com outro, *sobreposição*, pela aplicação de um operador.

2.2.4 Medição Quântica - Geral

Considere-se a seguinte transcrição de Nielsen and Chuang [2000][P.84]:
"Quantum Measurements are described by a collection M_m *of* measurement operators. *These are operators acting on the state space of the system being measured. The index* m *refers to the measurement outcomes that may occur in the experiment."*

Até ao momento, temo-nos concentrado em descrever ou enquadrar *o qubit* no modelo de mecânica quântica. Seguindo os postulados, começamos por descrever o *qubit*, o espaço vetorial bem como, a evolução do qubit. Em seguida, avançamos para a manipulação do estado, afirmando que qualquer operador unitário é capaz de alterar o vector estado de uma partícula. Durante este tempo, nada dissemos sobre como consultar a informação presente no estado, apenas procuramos deixar a intuição que, apesar de capacidade fenomenal do *qubit* armazenar informação, consultar a mesma, não seria uma tarefa trivial. No decorrer deste postulado e das próximas páginas, procuramos explicar como isto é feito e quais as limitações existentes.

Pelo enunciado do postulado, já percebemos que a operação de medição de um estado é consequência de um conjunto de *Operadores especiais*. Juntando o postulado anterior, afirmámos que em primeiro lugar, estes são operadores lineares unitários, como os que temos apresentado. E em segundo, que os mesmos modificam o estado do *qubit*. Considerando novamente o segundo postulado, é possível num intervalo de tempo, relacionar um estado anterior

com um posterior por intermédio da ação de um operador. Isto ocorre porque o operador modifica o estado. Concretizando isto para o *qubit*, e comparando-o com o seu *homólogo* clássico, vamos apresentar os possíveis *outputs* da *medição* de um estado, definido na base computacional.

1. Se o *qubit* $|\psi\rangle$, estiver em um dos dois estados base possíveis isto é, $|0\rangle$ ou $|1\rangle$, aquando medido/consultado iremos obter, respectivamente 0 ou 1. Este comportamento é expectável é coincide com o dos *bits* normais.

2. Se por outro lado o nosso *qubit* $|\psi\rangle$, estiver em sobreposição, onde α e β possuem um determinado valor, como o apresentado na equação 2.57 isto é, $\alpha = \beta = 1/\sqrt{2}$ então, **podemos obter ou $|0\rangle$ ou $|1\rangle$, com uma determinada probabilidade**.

Antes de esmiuçarmos o ponto 2, convém fazer um parêntesis sobre a abordagem clássica *versus* quântica. Na física clássica, os cientistas procuram determinar aquilo que é conhecido como, as variáveis *dinâmicas*[9] do sistema (para uma dada partícula). Como exemplo destas variáveis temos o *Momento Linear*, *Energia*, *Posição* ou mesmo *Velocidade*. E admitimos que quando estamos na posse destes fatores, podemos afirmar que "sabemos tudo o que há para saber" sobre a dita partícula. Em física quântica, muito devido ao facto da micro-escala em que os efeitos quânticos se manifestam[10] bem como, devido a nossa falta de compreensão de todos os fatores que podem contribuir/interferir com o estado, podemos apenas *prever*, qual o resultado mais provável de uma ação.

Posto isto, vamos introduzir a formulação dos operadores de medição expressos pelo postulado ao nível do resultado da medição obtida, assim como propriedades que estes devem respeitar. Considere-se o estado $|\psi\rangle$, para além disso e pelo enunciado do postulado, vamos considerar um operador de medição M_m sendo que o resultado da ação deste é espelhado por m. Assim sendo, se $|\psi\rangle$ for o estado do sistema antes de aplicarmos M, então a probabilidade de m acontecer é dada por:

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{2.59}$$

sendo o estado após a medição dado por,

$$\big|\Psi'\big\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}, \tag{2.60}$$

tal que o conjunto de operadores de medição, satisfazem, a seguinte equação,

$$\sum_m M_m^\dagger M_m = I. \tag{2.61}$$

[9] por comparação com as *estáticas.*, como a *Massa*

[10] abaixo do nivél do atmo

que por sua vez, implicitamente, garante que o somatório das probabilidades deve ser 1.

$$1 = \sum_m p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{2.62}$$

Com ajuda de um simples exemplo, vamos agora voltar ao ponto 2, efetuando uma operação de medição na base computacional. Seja M_0 e M_1 os respectivos operadores de medição na base $|0\rangle$ e $|1\rangle$ respectivamente,

$$M_0 = |0\rangle\langle 0| \tag{2.63}$$

$$M_1 = |1\rangle\langle 1|. \tag{2.64}$$

antes de mais, note-se como os operadores são ambos *Hermitianos* e que, $M_0^2 = M_0, M_1^2 = M_1$. Isto implica que se verifica a *equação de completude*, $I = M_0^\dagger M_0 + M_1^\dagger M_1 = M_0 + M_1$. Então no nosso estado $|\psi\rangle$, obtemos:

$$p(0) = \langle\psi|M_m^\dagger M_m|\psi\rangle = \langle\psi|M_0|\psi\rangle = \|a\|^2 \tag{2.65}$$

$$p(1) = \langle\psi|M_m^\dagger M_m|\psi\rangle = \langle\psi|M_1|\psi\rangle = \|b\|^2. \tag{2.66}$$

Ou seja, no caso especifico do nosso exemplo Equação 2.57, temos 50% de possibilidade, uma vez que $(1/\sqrt{2})^2 = 1/2$, de ao medir o estado obtermos $|0\rangle$ ou $|1\rangle$. Esse resultado é confirmado pelas seguintes equações:

$$\frac{M_0|\psi\rangle}{\|\alpha\|} = \frac{\alpha}{\|\alpha\|}|0\rangle \equiv |0\rangle \tag{2.67}$$

$$\frac{M_1|\psi\rangle}{\|\beta\|} = \frac{\beta}{\|\beta\|}|1\rangle \equiv |1\rangle. \tag{2.68}$$

Este fenómeno conhecido por *colapsamento* do estado, ocorre sempre que possuímos um estado em *sobreposição* e consultamos o seu valor. Isto é uma divergência importante dos *bits* clássicos, cujo estado podemos consultar a nosso belo prazer, sem que isso interfira com o estado do *bit*.
De seguida, outro aspecto pertinente sobre os estados quânticos é o da **distinção de estados não ortogonais**.

Considere-se o seguinte cenário. A *Alice* pretende definir um estado quântico $|\psi_i\rangle$ tal que, $1 \leq \psi_i \leq n$ enviando-o de seguida para o *Bob*. A finalidade desta tarefa é o *Bob* perceber o índice do estado que a *Alice* lhe enviou. Para isso o Bob podia definir um conjunto de operadores M_i,

$$M_i \equiv |\psi_i\rangle\langle\psi_i| \tag{2.69}$$

juntamente com o operador M_0. Desta forma o *Bob* teria um conjunto de operadores que cobriam os possíveis *i's*, e que satisfaziam a equação *completude*. Isto garantia que, se o estado $|\psi_i\rangle$ fosse preparado e sujeito ao operador M_i ou seja,

$$p(i) = \langle\psi_i|M_i|\psi_i\rangle = 1 \tag{2.70}$$

isso implicaria que o estado i era obtido com certeza isto é, com $prob = 1$. Tal só acontece se os estados $|\psi_i\rangle$ forem *ortogonais*. Considere-se os seguintes estados sujeitos ao operador de medição M_0:

$$M_0|0\rangle \Rightarrow |0\rangle, \text{ com 100\% de hipotesses} \tag{2.71}$$

$$M_0\frac{|0\rangle + |1\rangle}{\sqrt{2}} \Rightarrow |0\rangle, \text{com 50\% de hipotesses.} \tag{2.72}$$

Neste caso e intuitivamente reparamos que o *Bob* nada poderia afirmar sobre qual o estado $|\psi_i\rangle$. Pois ele poderia obter o resultado **0** das duas formas. Isto acontece porque os referidos estados não são ortogonais. Na seção do *produto interno*, referimos que a nossa sensibilidade para com o resultado da operação, $\langle\psi|\psi\rangle$ teria implicações mais tarde explicadas. A implicação é esta, se a operação do produto interno entre dois estados não for igual a zero, então isso implica que **é impossível distingui-los com total certeza**.

2.2.5 Medições Projectivas

Considere-se a seguinte transcrição de Nielsen and Chuang [2000][P.87]:
" A projective measurement is described by an observable, M, a Hermitian operator on the state space of the system being observerd".

Este tipo de medição é um caso particular do postulado geral. O postulado tira partido do fato do observável M possuir a seguinte *decomposição espectral*:

$$M = \sum_m mP_m. \tag{2.73}$$

onde P_m é um projetor para o *espaço próprio* de M, com o *valor próprio m*. Neste tipo particular de medição, o resultado obtido corresponde ao *valores próprios* do operador/projetor. Ou seja, considerando o estado $|\psi\rangle$, como o estado do sistema antes da medição, o postulado expressa a probabilidade de se obter m após a medição por:

$$p(m) = \langle\psi|P_m|\psi\rangle. \tag{2.74}$$

Sendo o estado $|\psi\rangle^{'}$ o estado resultante, dado por

$$|\psi\rangle^{'} = \frac{P_m|\psi\rangle}{\sqrt{p(m)}}. \tag{2.75}$$

Intuitivamente, conseguimos perceber que este operador, definido num dado espaço vectorial, projeta um determinado estado num *sub-espaço vectorial* do mesmo.[11] Pela sua definição (*Projector*), sabemos que a sua ação consecutiva se traduz no mesmo resultado. Mais ainda, pela Equação 2.75 podemos afirmar que o estado $|\psi\rangle$ é projetado no *sub-espaço* dado por m, com probabilidade proporcional à sua magnitude.[12] Isto implica que, quando esse

[11] vetor próprio do operador.
[12] raiz quadrada do comprimento do vector

sub-espaço tem dimensão 1, corresponde à *medição geral, com respeito a base ortonormal, que vimos no postulado anterior.*

Este tipo de medição apresenta muitas propriedades interessantes. Nomeadamente, permite-nos facilmente calcular o *valor médio* para *medições projectivas.* Este valor médio associado a um operador observável é dado por:

$$\langle\psi|M|\psi\rangle, \tag{2.76}$$

e não deve ser confundido com o valor mais provável de ocorrer. Esta operação, cuja sintaxe se encontra na tabela 2.1, **exprime o significado estatístico**(do ponto de vista quântico), do conjunto de resultados obtidos nas experiências efetuadas.[13]

Vamos de seguida apresentar um exemplo de *medições projectivas.* Considere-se o observável:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.77}$$

Podemos representar o mesmo segundo a sua *decomposição espectral*:

$$Z \equiv |0\rangle\langle 0| - |1\rangle\langle 1|. \tag{2.78}$$

Recorrendo ao *SAGE*, podemos facilmente comprovar que este observável, tem *valores próprios* +1 e -1, com o respectivos *vetores próprios* $|0\rangle$ e $|1\rangle$. Posto isto, se considerarmos como *input* o estado:

$$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \tag{2.79}$$

verificamos que, a medição de $\mathbf{Z}$ em $|\psi\rangle$ devolve o resultado +1 com probabilidade $\langle\psi||0\rangle\langle 0||\psi\rangle = 1/2$ e respectivamente o resultado -1, com probabilidade 1/2. Além disso, se considerarmos um *estado/registro* com dois elementos ou *qubits* isto é,

$$|\varphi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{2.80}$$

Bem como os dois projetores dados por, $P_0 \otimes 1$ e e $P_1 \otimes 1$, ou seja

$$P_0 = |01\rangle\langle 01|; \tag{2.81}$$

$$P_1 = |11\rangle\langle 11|. \tag{2.82}$$

verificamos que a informação em *sobreposição* não é completamente destruída, uma vez que ao realizar a medição obtemos respectivamente:

$$P_0|\varphi\rangle \equiv \alpha_{00}|00\rangle + \alpha_{01}|01\rangle; \tag{2.83}$$

$$P_1|\varphi\rangle \equiv \alpha_{10}|10\rangle + \alpha_{11}|11\rangle. \tag{2.84}$$

Ou seja, sempre que a dimensão do *espaço próprio* de um determinado projetor P_i é maior que 1, então a operação de medição *preserva* "alguma" informação particular dessa sobreposição no *sub-espaço* dado pela projeção.

[13] para uma grande repetição das mesmas.

2.2.6 Medições POVM

As medições POVM - *positive-operator valued measure*, surgem no estudo das probabilidades de determinadas medições ocorrerem. Estas contrastam com aquelas onde o principal interesse é o estado do sistema pós-medição. Isto é particularmente útil quando numa experiência medimos o sistema apenas uma vez, normalmente no fim da experiência. Concretamente, dado uma coleção de operadores positivos[14] E_j que satisfazem a seguinte equação,

$$\sum_j E_j = I, \tag{2.85}$$

pretendemos com os mesmos, medir um determinado sistema quântico. Pela definição das medições **POVM**, a probabilidade do resultado **j** ocorrer é dada por,

$$p(j) = \langle\psi|E_j|\psi\rangle. \tag{2.86}$$

Vamos concretizar estes conceitos com um pequeno exemplo. Seja $|\psi\rangle = |0\rangle$ o vector, que descreve o nosso sistema quântico. Considere-se também o seguinte conjunto de operadores **POVM** $\{E_1, E_2, E_3\}$[15], dados por:

$$\begin{cases} E_1 \equiv \frac{\sqrt{2}}{1+\sqrt{2}}|1\rangle\langle 1|, \\ E_2 \equiv \frac{\sqrt{2}}{1+\sqrt{2}}\frac{(|0\rangle-|1\rangle)(\langle 0|-\langle 1|)}{2}, \\ E_3 \equiv I - E_1 - E_2. \end{cases} \tag{2.87}$$

tal que,

$$\sum_{j=1}^{3} E_j = I. \tag{2.88}$$

Com estes operadores as seguintes diferentes probabilidades p(i) imperam, quando aplicamos os mesmos ao estado $|\psi\rangle$.

$$\begin{aligned} p(1) &= \langle\psi|E_1|\psi\rangle = 0 \\ p(2) &= \langle\psi|E_2|\psi\rangle = \frac{\sqrt{2}}{[2(1+\sqrt{2})]} \\ p(3) &= \langle\psi|E_3|\psi\rangle = \frac{(2+\sqrt{2})}{(2+2\sqrt{2})}. \end{aligned} \tag{2.89}$$

Vamos finalizar esta seção refinando o exemplo apresentado, procurando com isto explicar a futura aplicação deste tipo de medição, num cenário

[14] Qualquer $v \in V$(espaço vectorial). t.q $E \in V$, E é um operador positvo ($E > 0$) se $\langle E(v)|v\rangle > 0$

[15] retirados de Nielsen and Chuang [2000]

concreto. Seja $|\phi\rangle$ o seguinte estado,

$$|\phi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \tag{2.90}$$

considerem-se os seguintes intervenientes numa conversa quântica, Alice e Bob. Supondo que a Alice envia ao Bob o *qubit* $|\phi\rangle$ ou $|\psi\rangle$, este não possui nenhuma forma de os distinguir com total certeza, como já explicamos. No entanto recorrendo aos operadores E_1, E_2, E_3 o Bob consegue ganhar alguma informação sobre o estado. **Concretamente, ele consegue distinguir os estados algumas vezes, no entanto nunca erra**.
Supondo que o Bob recebe o estado $|\psi\rangle = |0\rangle$, ele executa a medição **POVM** dada por, $\{E_1, E_2, E_3\}$. A probabilidade deste observar o resultado E_1 é zero, uma vez que $\mathrm{p}(1) = 0$. Ou seja, se esta foi a probabilidade obtida, o Bob consegue inferir que o estado que ele recebeu deve ter sido $|\phi\rangle$. A mesma linha de raciocínio é usada se o Bob obtiver E_2, neste caso ele infere que deve ter recebido $|\psi\rangle$. No entanto algumas vezes ele obtém E_3. Nestes casos ele não consegue inferir nada sobre o estado recebido. O estado perde-se!

2.2.7 Fase Qântica

Fase é um termo da *framework* da mecânica quântica. Como tal, ela aparece várias vezes associada ao estado quântico. No entanto, esta possui diferentes significados consoante o contexto. Nas próximas linhas procuramos rever estas diferenças, na utilização da palavra *fase*. *Fase* não é mais que uma quantidade um fator, que adicionamos a um estado. A questão é em que medida ela o modifica. Vamos começar por apresentar os dois diferentes contextos em que para nós, se apresenta o conceito de fase. Existe a *fase relativa* e a *fase global*. Começando pela segunda, vamos considerar o estado:

$$e^{i\theta}|\psi\rangle, \tag{2.91}$$

em que $|\psi\rangle$ é o vector estado e θ é um numero real. Neste caso, dizemos que o estado $|\psi\rangle$ é igual ao estado em cima apresentado, *a menos de uma diferança fase*, dada pelo *fator de fase global* $e^{i\theta}$. Experimentalmente, observou-se que o valor esperado ou as estatísticas de medição destes dois estados são as mesmas. Considere-se M_m um operador de medição associado a um qualquer estado quântico. Ou seja,

$$\langle\psi|M_m^\dagger M_m|\psi\rangle \equiv \langle\psi|e^{-i\theta}M_m^\dagger M_m e^{i\theta}|\psi\rangle = \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{2.92}$$

Isto é, **do ponto de vista de quem observa, os resultados são iguais**!!
Por outro lado na *fase relativa* já não é a mesma coisa. Considere-se os dois estados dados por

$$\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \qquad\qquad \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{2.93}$$

olhando para as amplitudes do estados $|1\rangle$, verificamos que estas diferem em *sinal*, sendo iguais em *magnitude*, com valor $\frac{1}{\sqrt{2}}$. Neste caso, dizemos que duas amplitudes α e β diferem *por um factor de fase relativa* se existir um numero real θ tal que, $\alpha = e^{i\theta}\beta$. No nosso exemplo, verificamos que as amplitudes para $|0\rangle$ são idênticas (*fator fase relativa de 1*) e as amplitudes de $|1\rangle$, diferem por um *fator de fase relativa* igual a -1. Concluindo, a diferença entre *fase relativa* e *fase global* é que na relativa as diferenças podem variar de amplitude para amplitude, enquanto que na *global* são idênticas.

2.2.8 Sistemas Compostos

Considere-se a seguinte transcrição de Nielsen and Chuang [2000][P.94]:
"The Hilbert space describing a composite physical system is the tensor product of the spaces describing the individual systems, and the state vector of the system is the tensor product of the individual state vectors. A quantum register is the tensor product of individual qubits."

Até ao momento apenas descrevemos sistemas considerados simples. Analisando o nosso caso de interesse - computação quântica - todos os conceitos ou operadores que analisamos, estavam definidos para apenas *1-qubit*. Da mesma forma que num computador clássico, um *bit* não nos leva muito longe, o mesmo sucede em computação quântica. Precisamos por isso de emparelhar *qubits*, para obtermos computação. Pelo enunciado do postulado, verificamos que isto é conseguido recorrendo a operação de *Produto Tensorial*. Considere-se os dois seguintes *qubits*

$$\begin{aligned} |0\rangle &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ |1\rangle &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \tag{2.94}$$

Como sabemos eles representam a base computacional do nosso sistema. Sabemos também, que eles formam uma *base* ortogonal para o espaço de *Hilbert* onde definimos os nossos problemas. Pretendemos que ao aumentar este espaço vectorial, todos os conceitos e propriedades sejam preservados. Vamos agora observar o que acontece ao realizarmos o produto tensorial entre eles:

$$|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \equiv |01\rangle. \tag{2.95}$$

É fácil perceber que a *base* do espaço vectorial resultante vai ser de maior dimensão, de maneira acompanhar o crescimento do espaço. Relembrando o Capítulo 2.1.1 sobre a *bases* de um espaço vectorial, facilmente a rescrevemos

como:

$$|0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \equiv |00\rangle; \tag{2.96}$$

$$|1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \equiv |10\rangle; \tag{2.97}$$

$$|1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \equiv |11\rangle. \tag{2.98}$$

Refletindo isto, o *qubit* passa então a ser definindo recorrendo a *quatro amplitudes*(α_{nn}), ao invés de apenas duas (α e β). Ou seja,

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{2.99}$$

sujeito a condição de normalização,

$$\|\alpha_{00}\|^2 + \|\alpha_{01}\|^2 + \|\alpha_{10}\|^2 + \|\alpha_{10}\|^2 = 1. \tag{2.100}$$

Neste momento e pelo enunciado do postulado, definimos um *registo* como o *vector* ou *estado*, $|\rangle$, resultante da operação de tensor entre múltiplos *qubits*. E quanto aos *operadores* que aqui temos descrito? Como analisamos na seção dedicada ao produto tensorial, o mesmo podia ser aplicado a matrizes bem como vectores. Naturalmente surge então que, dado um operador $\mathbf{Q}$ que atue num sistema de um único *qubit*, o mesmo pode ser rescrito para atuar num *registo* de *n qubits*, por

$$Q^{\otimes n}. \tag{2.101}$$

Vamos considerar um exemplo para perceber como isto acontece na prática. Considere-se os dois *qubits*, $|0\rangle$ e $|1\rangle$, bem como os operador $\mathbf{H}$, tal que,

$$H|0\rangle \equiv \frac{(|0\rangle + |1\rangle)}{\sqrt{2}}; \qquad H|1\rangle \equiv \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}. \tag{2.102}$$

Se quisermos usar o mesmo operador $\mathbf{H}$, mas agora aplicada a um *registo* de 2 *qubits* na nossa *base computacional*, então precisamos de calcular $H^{\otimes 2}$:

$$H^{\otimes 2} \equiv \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}. \tag{2.103}$$

Podemos então rapidamente calcular os 4 possíveis resultados deste operador para a nova base computacional,

$$|\psi_{00}\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}; \tag{2.104}$$

$$|\psi_{01}\rangle = \frac{|00\rangle - |01\rangle + |10\rangle - |11\rangle}{2}; \tag{2.105}$$

$$|\psi_{10}\rangle = \frac{|00\rangle + |01\rangle - |10\rangle - |11\rangle}{2}; \tag{2.106}$$

$$|\psi_{11}\rangle = \frac{|00\rangle - |01\rangle - |10\rangle + |11\rangle}{2}. \tag{2.107}$$

Note-se que, as colunas da matriz refletem cada uma das transformações efetuadas ao estado. Pretendemos agora generalizar este resultado. Considere-se a seguinte aplicação do operador *Hadamard* a um estado genérico $|\phi\rangle$.

$$H|\phi\rangle = \sum_{k=0}^{1} \frac{(-1)^{\langle\phi|k\rangle}|k\rangle}{\sqrt{2}}. \tag{2.108}$$

Comparando com a equação 2.102, vemos que esta se trata de uma generalização, aplicada a um *qubit*. Uma vez que $-1^{\langle\phi|k\rangle}$ ou é -1 ou 1. Generalizando agora para um sistema de *n-qubits* obtemos,

$$H^{\otimes n}|\phi_1, \phi_2, \ldots, \phi_n\rangle = \sum_{k_1,\ldots,k_n} \frac{(-1)^{\phi_1 k_1 + \phi_2 k_2 + \ldots + \phi_n k_n}|k_1, k_2, \ldots, k_n\rangle}{\sqrt{2^n}}, \tag{2.109}$$

que pode ser simplificada como,

$$H^{\otimes n}|\phi\rangle = \sum_{k} \frac{(-1)^{\langle\phi|k\rangle}|k\rangle}{\sqrt{2^n}}. \tag{2.110}$$

Um resultado importante para nós, dedutível da simplificação em cima apresentada é o seguinte,

$$H^{\otimes n}|0\rangle^{\otimes n} = \sum_{k} \frac{|k\rangle}{\sqrt{2^n}}, \tag{2.111}$$

onde $|\phi\rangle = |0\rangle^{\otimes n}$ e $\langle\phi|k\rangle = 0$, qualquer que seja k.

2.3 Sumário

Durante este capítulo, tivemos oportunidade de rever alguns conceitos de álgebra linear, enquadrando os mesmos nas nossas necessidades bem como, dar a conhecer a sintaxe em uso no modelo de computação quântica. Apresentamos também um conjunto de postulados cuja finalidade é estabelecer a ponte entre o modelo matemático e a sua aplicação num sistema físico concreto. Teremos a oportunidade de os constatar nas próximas páginas do documento.

No próximo capítulo, avançaremos para o tópico *circuitos quânticos*, onde descreveremos as propriedades e funcionamento do modelo computacional quântico.

Capítulo 3

Circuitos Quânticos

3.1 Circuitos Quânticos

Até ao momento, procurou-se cobrir a base matemática que apoia o modelo de computação quântica, bem como introduzir o elemento basilar deste novo modelo computacional, o *qubit*. Durante essa fase foram introduzidos conceitos de álgebra linear e numa fase posterior, uma série de postulados cuja finalidade prende-se com a ligação entre o modelo abstrato, puramente matemático, e a realização deste no mundo físico.

Neste capítulo centra-se em dois objectivos claros. Primeiro, procura detalhar o funcionamento lógico por detrás deste novo modelo de computação, realizável através de *circuitos quânticos*. Em segundo lugar, explica que existe um conjunto muito reduzido de operadores lógicos[1] capazes de oferecer universalidade, tanto do ponto de vista clássico como quântico.

Da computação clássica sabemos que os *bits*, responsáveis por armazenar a informação num computador, são modificados através da aplicação estruturada de um conjunto de operadores lógicos. Como exemplo, temos o *not*, *and*, *xor*, entre outros, cujo comportamento é definido recorrendo a tabelas de verdade. Esta rede de operadores lógicos forma um circuito, que dado um determinado *input*, vai produzir o desejado *output*. Analogamente, um circuito quântico segue a mesma lógica. No entanto distingue-se dos clássicos em alguns pontos. Dois dos mais importantes são:

[1] do inglês "logic gates"

- As operações são reversíveis isto é, a aplicação de um operador não destrói o seu *input*, como acontece em alguns operadores clássicos.

- Os circuitos quânticos, permitem acelerar alguns problemas computacionais, atingindo reduções de tempo de nível exponencial. Sendo este um dos principais motivos para o estudo desta área.

As próximas páginas estarão divididas por diversos capítulos. Começaremos por introduzir alguns dos mais famosos operadores bem como, analisar o comportamento destes no elemento mais básico da computação quântica - o *qubit*. Iremos mesmo mais além, não nos ficando apenas pela análise matemática da operação, mas pela visualização destas modificações na esfera de *Bloch*, recorrendo a *scripts* desenvolvidos em **Python/SAGE**. Ainda neste capítulo, abordaremos outras operações pertinentes sobre um *qubit*, tirando sempre que possível partido da plataforma SAGE como temos vindo a fazer.

De seguida avançaremos para *operações controladas*, onde se analisarão circuitos para um sistema de múltiplos *qubits*, uma vez que, à semelhança da computação clássica, realizar computação com um único *bit* não nos leva muito longe. Posteriormente, mas ainda dentro desse capítulo analisaremos dois algoritmos conhecidos por *gray-codes* e *two-level unitaries*, que decompõem um operador que atua num sistema de n *qubits*, num conjunto de operadores que atuam por si só apenas num *qubit* desse sistema.

Estes dois resultados, vão-nos permitir concluir sobre a seção seguinte, onde abordaremos a questão sobre a universalidade de um conjunto muito particular de operadores lógicos.

3.2 Operações num qubit

Nesta seção, pretende-se apresentar os principais operadores lógicos existentes, tendo como objetivo prático, a sua realização quântica, num sistema de apenas um *qubit*. Faz por isso sentido recapitular muito rapidamente tudo aquilo que aprendemos sobre esta nova entidade. Como vimos no capítulo anterior, um *qubit* é um vector, cujo estado é dado por:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.1}$$

onde α e $\beta \in \mathbb{C}$ e são parametrizáveis, estando sujeitos à seguinte condição de normalização:

$$\|\alpha\|^2 + \|\beta\|^2 = 1. \tag{3.2}$$

Mais ainda, verificamos que as operações num *qubit* devem preservar a equação de normalização do estado, pelo que são descritas através de matrizes unitárias 2×2. A seguinte discussão baseia-se no trabalho de Glendinning [2005].

O estado de um *qubit* ou melhor, os estados possíveis que um *qubit* pode atravessar, podem ser interpretados geometricamente como o conjunto de pontos na superfície de uma esfera de raio 1 (equação de normalização). Essa esfera dá pelo nome de *esfera de Bloch* e pode ser vista como uma generalização da representação de um número complexo, $\|z\|^2 = 1$, num círculo unitário. A sua representação geométrica é visível na Figura 3.1.

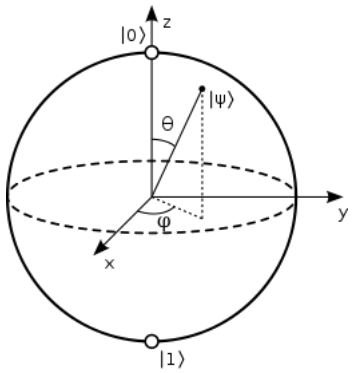


Figura 3.1: *Esfera de Bloch* - $\Psi = \cos\frac{\theta}{2}|0\rangle + e^{i\times\varphi}\sin\frac{\theta}{2}|1\rangle$

No entanto a representação de um estado na *esfera Bloch* não é imediata. Como já constatamos, um estado é representado por um vector. A conversão deste para um ponto na superfície da esfera envolve três passos. Primeiro, escrevemos o estado em coordenadas polares:

$$|\Psi\rangle = r_\alpha e^{i\varphi_\alpha}|0\rangle + r_\beta e^{i\varphi_\beta}|1\rangle. \tag{3.3}$$

Em seguida aplicamos uma fase global, $e^{-i\varphi_\alpha}$, que sabemos não ter efeitos observáveis:

$$|\Psi\rangle = r_\alpha|0\rangle + r_\beta e^{i\varphi}|1\rangle \text{ onde, } \varphi = (\varphi_\beta - \varphi_\alpha) \tag{3.4}$$

Voltando a passar o coeficiente de $|1\rangle$ para coordenadas cartesianas e adicionado o que sabemos sobre a equação de normalização, obtemos:

$$\|r_\alpha\|^2 + \|x + iy\|^2 = 1 \equiv r_\alpha^2 + x^2 + y^2 = 1 \tag{3.5}$$

Neste momento já temos uma equação de uma esfera expressa em três coordenadas (x,y,r_α). O último passo consiste em tirar partido da relação que existe entre as coordenadas cartesianas e esféricas.

$$x = r\sin\theta'\cos\varphi \tag{3.6}$$

$$y = r\sin\theta'\sin\varphi \tag{3.7}$$

$$z = r\cos\theta'. \tag{3.8}$$

Sabendo que o raio da esfera é 1 e considerando $z = r_\alpha$, então o estado passa a ser definido por:

$$|\Psi\rangle = \cos\theta'|0\rangle + e^{i\varphi}\sin\theta'|1\rangle \tag{3.9}$$

Falta agora apenas especificar, quais os ângulos a considerar para conseguirmos varrer todos os pontos da esfera. Note-se que $\theta^{'} = \arccos(r_\alpha)$ que se encontra sempre no intervalo $0 \leq \theta^{'} \leq \frac{\pi}{2}$(porque, pela aplicação da fase global, $r_\alpha \geq 0$). Por outro lado, observa-se que todos os pontos da equador (i.e $\theta^{'} = \frac{\pi}{2}$) podem ser identificados (só diferem de um factor de fase global). Assim, fazendo:

$$\theta = 2\theta^{'}. \tag{3.10}$$

podemos *mapear* todos os pontos do hemisfério superior, inclusive aqueles que se situavam no equador da semi-esfera,[2] na globalidade da esfera. Ou seja, obtemos agora uma esfera dada por,

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \text{ onde, } 0 \leq \theta \leq \pi, 0 \leq \varphi \leq 2\pi \tag{3.11}$$

Como é possível observar, voltamos a conseguir representar um estado apenas com duas coordenadas, mas desta vez na superfície de uma esfera. Isto vai ser muito útil para perceber graficamente o que acontece, quando aplicamos um operador a um estado, já que esta ação será traduzida numa rotação de um ponto (estado) na esfera.

As matrizes *Pauli* são três dos operadores mais conhecidos e a sua representação matricial é:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{3.12}$$

Como explicado no Capítulo 2, o Teorema da Decomposição Espectral (teorema 1 - P. 16) permite-nos exprimir cada um destes operadores como, $A = \sum_a a|a\rangle\langle a|$. Considere-se a seguinte função,

$$f :: C \to C$$
$$f(A) \equiv \sum_a f(a)|a\rangle\langle a| \tag{3.13}$$

Esta metodologia permite-nos definir um conjunto de operações nossas conhecidas aplicadas agora a operadores unitários. Como exemplo podemos ter, a raiz quadrada de um operador, o logaritmo, ou então a exponenciação, representadas pela definição da função $\mho$. Considere-se $exp(ix\sigma)$, para um dado número *real* x e para uma matriz σ, tal que, $\sigma^2 = \mathrm{I}$. Então se realizarmos a *expansão* da exponencial obtemos

.

$$e^{ix\sigma} = \sum_{n=0}^{\infty} \frac{(ix\sigma^n)}{n!}. \tag{3.14}$$

[2] dados por, $e^{i\theta}|1\rangle$

dividindo a soma, nas suas componentes *pares* e *ímpares*.

$$e^{ix\sigma} = \sum_{n-par} \frac{(ix\sigma)^2}{n!} + \sum_{m-impar} \frac{(ix\sigma)^2}{m!}. \tag{3.15}$$

Como sabemos $\sigma^2 = I$. Isto permite-nos inferir que quando σ é exponenciado a uma *potência par* o resultado será I. Por outro lado se o expoente for *impar* o resultado será σ. Tirando partido disto,

$$e^{ix\sigma} = \sum_{n-par} \frac{(i)^n x^n}{n!} + \sum_{n-impar} \frac{(i)^m x^m \sigma}{m!}. \tag{3.16}$$

Se agora considerarmos $n = 2k$ e $m = 2l+1$, podemos rescrever os somatórios como,

$$e^{ix\sigma} = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{2k!} + i\sigma \sum_{l=0}^{\infty} \frac{(-1)^l x^{2l+1}}{(2l+1)!}. \tag{3.17}$$

Analisando este resultado, verificamos que as duas partes da soma são na verdade a expansão das séries de *Taylor*, do **seno** e **co-seno** ou seja

$$e^{ix\sigma} = \cos x + i\sigma \sin x. \tag{3.18}$$

□

Se generalizarmos o resultado apresentado para um dado vetor $n = (n_x, n_y, n_z)$ tal que, este é um vector unitário, então podemos redefinir a equação de rotação sobre o referido vetor como,

$$R_n(\theta) \equiv e^{-i\theta n.\frac{\sigma}{2}} = \cos\left(\frac{\theta}{2}\right) - i(n_x X + n_y Y + n_z Z)\sin\left(\frac{\theta}{2}\right) \tag{3.19}$$

onde σ, corresponde ao vector **(X, Y, Z)** com as matrizes *Pauli*. Apresenta-se de seguida o resultado da exponenciação das matrizes *Pauli*.

$$R_x(\theta) \equiv e^{-i\theta\frac{X}{2}} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}X = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{3.20}$$

$$R_y(\theta) \equiv e^{-i\theta\frac{Y}{2}} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Y = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{3.21}$$

$$R_z(\theta) \equiv e^{-i\theta\frac{Z}{2}} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}Z = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{3.22}$$

A exponenciação das matrizes *Pauli*, geometricamente equivalem a rotações em torno dos eixos **x, y, z**. Esta propriedade tem diversas utilidades. A título de exemplo, a aplicação do operador **X** ao estado $|0\rangle$ teria neste o efeito da operação *not* ou seja,

$$|0\rangle \equiv \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.23}$$

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \equiv |1\rangle \tag{3.24}$$

Isto pode ser obtido fazendo uma rotação do estado. Considerando agora $R_x(\pi)$ e o estado $|0\rangle$ vemos que obtemos o estado $|1\rangle$ sujeito a uma diferença de fase.

$$|0\rangle \equiv \begin{bmatrix}1\\0\end{bmatrix} \Rightarrow \begin{bmatrix}\cos(0)\\0\end{bmatrix} \tag{3.25}$$

$$R_x(\pi)|0\rangle = \begin{bmatrix}0 & -i\\-i & 0\end{bmatrix} \times \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\-i\end{bmatrix} \equiv -i\begin{bmatrix}0\\1\end{bmatrix} \equiv |1\rangle \tag{3.26}$$

Mais ainda, uma rotação permite-nos colocar um estado base em sobreposição. Consideremos agora na mesma o estado $|0\rangle$ bem como, o operador $R_x(\theta)$ sendo $\theta = \frac{\pi}{2}$:

$$|0\rangle \equiv \begin{bmatrix}1\\0\end{bmatrix} \Rightarrow \begin{bmatrix}\cos(0)\\0\end{bmatrix} \tag{3.27}$$

$$R_x(\frac{\pi}{2})|0\rangle = \begin{bmatrix}\cos\frac{\pi}{4} & -i\sin\frac{\pi}{4}\\-i\sin\frac{\pi}{4} & \cos\frac{\pi}{4}\end{bmatrix} \times \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}}\\-i\frac{1}{\sqrt{2}}\end{bmatrix} \equiv \frac{|0\rangle + i|1\rangle}{\sqrt{2}} \tag{3.28}$$

Como é visível, com apenas uma rotação colocamos um estado base em sobreposição. De seguida veremos como esta ação é traduzida do ponto de vista de um circuito quântico.

$$|0\rangle \text{ ——— } \boxed{R_x(\pi)} \text{ ——— } \frac{|0\rangle+i|1\rangle}{\sqrt{2}} \tag{3.29}$$

Ao conjunto de operadores em cima apresentados, acrescentamos mais três vulgarmente utilizados, nomeadamente o operador *Hadamard* (conhecido por H), *Fase* (conhecido por S) e $\frac{\pi}{8}$ (denominado de T);

$$H \equiv \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}; S \equiv \begin{bmatrix}1 & 0\\0 & i\end{bmatrix}; T \equiv \begin{bmatrix}1 & 0\\0 & e^{i\frac{\pi}{4}}\end{bmatrix}. \tag{3.30}$$

Analisando o comportamento destes como já foi explicado isto é, olhando para as colunas, que refletem a ação do operador em cada elemento da base computacional, verificamos:

- O operador *Hadamard*, desloca cada um dos estados base $|0\rangle$ e $|1\rangle$ para um estado *intermédio* entre os mesmos, isto é :
  1. $H|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$
  2. $H|1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$

$$\alpha|0\rangle + \beta|1\rangle \text{ ——— } \boxed{H} \text{ ——— } \frac{\alpha|0\rangle+|1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Figura 3.2: *Aplicação do operador* Hadamard *a um único qubit*

3. Este operador é muitas vezes denotado como a raiz-quadrada do operador **NOT**.

4. No anexo D.3, seção D.3.1 é possível consultar a animação desta operação, bem como os "scripts" que deram origem à mesma. Para efeitos desta, recorreu-se a utilização da *framework* - *Qutip*[3].

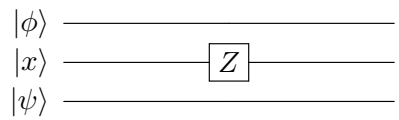
- O operador **S**, coloca uma diferença de fase **i** no elemento $|1\rangle$ deixando o $|0\rangle$ inalterado.

Figura 3.3: *Aplicação do operador de* fase *a um único qubit*

- A sua transformação na esfera de Bloch pode ser observada no anexo D.3, seção D.3.2.

- Por último o operador **T**, tem um comportamento semelhante ao operador **S**, deferindo apenas no valor que é acrescentado ao componente $|1\rangle$, que passa a ser $e^{i\frac{\pi}{4}}$

Até o momento apenas descrevemos circuitos simples. A título introdutório, vamos de seguida introduzir um circuito mais genérico.



(3.31)

Como é possível observar, o *input* deste circuito é composto por três *qubits* diferentes. Globalmente, o estado de *input* é dado pela operação do *produto tensorial* entre os seus componentes, respectivamente, $|\phi\rangle \otimes |x\rangle \otimes |\psi\rangle$. Analisando agora o circuito, verificamos que o primeiro e o último *qubit* não sofrem qualquer transformação entre o *input* e o *output*. O único que é transformado é o segundo *qubit* pela aplicação do operador Z. Estamos agora em condições de construir o operador que traduz esta transformação global. Para o fazermos, basta realizar o produto tensorial entre os operadores presentes, pela ordem descrita isto é, $I_{2\times 2} \otimes Z \otimes I_{2\times 2}$,

[3] `http://code.google.com/p/qutip/`

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix} \tag{3.32}$$

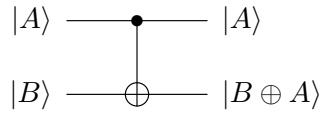
3.3 Multiplos Qubits - Operações Controladas

Como o nome indica, esta seção vai incidir sobre operações que afetam múltiplos *qubits*. O cerne deste capítulo é então descrever como controlar a afetação de múltiplos *qubits*. A forma mais simples de entender o conceito é pensar na instrução clássica "If". Concretamente, todos compreendemos o significado da seguinte expressão, "If A is true, then do B". Em computação quântica existe uma sintaxe equivalente. Nela definem-se dois tipos de *qubits* respectivamente, *qubit de controlo* e *qubit alvo*. Sobre os primeiros é testada a condição ou seja "se *os qubit de controlo*", afetando em caso de veracidade os demais *qubit* (*qubit alvo*). Um dos mais conhecidos e mais simples exemplos de *Operações Controladdas* é o *Not controlado* ou **CNOT**(do inglês). Neste caso, existem dois *qubits* tal que, se o *qubit* de *controlo* tiver o valor **1**, é efetuada a troca do *qubit alvo*, como se observa de seguida.

Control	Alvo	Saida
0	0	0
0	1	1
1	0	1
1	1	0

(3.33)

Esta operação pode ser vista como uma generalização do operador clássico **XOR**. Assim sendo, podemos então considerar o operador **CNOT** como um *XOR* controlado cuja representação num circuito quântico é:



(3.34)

Com a informação da tabela, bem como do circuito, podemos construir o operador CNOT da perspectiva matricial. Mais uma vez, observa-se o que acontece a cada um dos possíveis estados $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$. Apresenta-se de seguida o resultado da operação CNOT, salientando-se a sua ação em cada coluna da seguinte matriz:

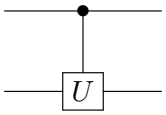
$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.35}$$

Figura 3.4: Duas representações possíveis para a operação *NOT-controlada*

Este exemplo é dos mais simplistas no que toca a operações controladas. Vamos agora analisar a formulação deste tipo de construção,

$$|c\rangle|t\rangle \longrightarrow |c\rangle U^c|t\rangle \tag{3.36}$$

que na nossa base computacional, se traduz em apenas aplicar o operador U ao *qubit* alvo, se o *qubit* de controlo tiver o valor 1 ($\equiv |1\rangle$). Caso contrário a entrada não é alterada. Isto é facilmente seguido na "representação" lógica do circuito apresentado em cima. Apresentamos de seguida a construção geral de um circuito para uma operação controlada com um *qubit* de controlo.



(3.37)

Antes de mais, interessa mostrar as seguintes equivalências entre circuitos, respectivamente:

$$\begin{aligned} &HXH = Z; \quad HYH = -Y; \quad HZH = X; \\ &XYX = -Y; \quad XZX = -Z; \quad T = e^{i\frac{\pi}{8}} e^{-i\frac{\pi}{8}} Z. \end{aligned} \tag{3.38}$$

Estas são facilmente demonstradas, realizando para tal a multiplicação entre os operadores. Pretende-se agora ilustrar como é possível construir um qualquer unitário, recorrendo a uma operação controlada. Para isto precisamos de introduzir o teorema *Z-Y decomposição para um único qubit.*

Teorema 2. (Z-Y decomposição para um único qubit) - Supondo que U é uma operação unitária num único qubit. Então existem quatro números reais, α, β, γ e δ tais que,

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{3.39}$$

Se definirmos quaisquer m e n como dois vectores reais não paralelos, cuja norma é 1 (unitários), podemos rescrever o teorema por,

$$U = e^{i\alpha} R_n(\beta) R_m(\gamma) R_n(\delta). \tag{3.40}$$

Este resultado vai-nos permitir mais tarde inferir que é possível exprimir um operador como um produto de rotações em dois eixos m e n, desde que os mesmos não sejam paralelos. A utilidade do Teorema 2 é expressa no seguinte corolário:

Corolario 1. Supondo que U é um operador unitário que atua num único qubit. Existem três operadores unitários A,B,C, que actuam num único qubit, tal que, ABC = I e U = $e^{i\alpha}$AXBXC, em que α, representa o fator da fase global.

A prova destes resultado pode ser consultada em Nielsen and Chuang [2000](pag. 176). O mesmo é fundamental para a construção de operações unitárias em *multi-qubits*. Como exemplo e recorrendo aos resultados da prova do Teorema, aplicou-se o Teorema para decompor o operador H, obtendo-se os valores de $\alpha, \beta, \delta, \lambda$, respectivamente, $\alpha = \pi$, $\delta = \pi$, $\gamma = -\frac{\pi}{2}$, $\beta = 0$(Anexo C.2). Posto isto, usando agora o resultado da prova do corolário presente na mesma página, construiu-se os seguintes operadores dados por,

$$A \equiv R_z(\beta)R_y(\frac{\gamma}{2}) \tag{3.41}$$

$$B \equiv R_y(-\frac{\gamma}{2})R_z(-\frac{(\gamma+\beta)}{2}) \tag{3.42}$$

$$C \equiv R_z(\frac{(\delta+\beta)}{2}) \tag{3.43}$$

sujeitos a,

$$U = e^{i\alpha}AXBXC \tag{3.44}$$

$$ABC = I. \tag{3.45}$$

Em seguida apresentamos o circuito que traduz este Teorema. Como se observa na Figura 3.5 estão presentes dois *fios quânticos* responsáveis por transmitir 2 *qubits* em simultâneo. Estes dois *qubits*, são vistos como *controlo* e *alvo* respectivamente e traduzem uma determinada ação ao nível do *qubit alvo*, que só é aplicada caso o *qubit de controlo* o especifique. Continuando a nossa análise, percebemos que o circuito faz exatamente o que suposto uma vez que, se o *qubit de controlo* for $|1\rangle$ então a operação $U = e^{i\alpha}AXBXC$ é aplicada. Por outro lado se o *qubit de controlo* for $|0\rangle$, então $ABC = I$ e nenhuma alteração é feita ao estado. Então podemos afirmar que o circuito apresentado implementa uma *operação controlada* num único *qubit*.

Nos exemplos observados até agora, todas as operações controladas dependiam do *qubit de controlo* estar programado para 1. Como é de esperar, deve também haver forma de controlar uma operação estipulando ao invés

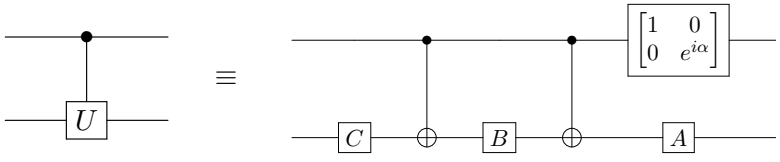


Figura 3.5: Circuito que implementa o operador *Hadmard* controlado com A,B,C e α, respeitando $U = e^{i\alpha}AXBXC, ABC = I$

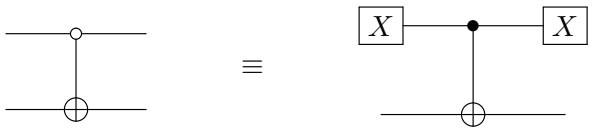


Figura 3.6: Operação controlada com o operador *NOT* a ser aplicado ao segundo qubit, condicionado ao primeiro ter o valor zero.

o *qubit de controlo* como zero. Considerando um exemplo simples, veremos como isto pode ser feito bem como a sintaxe que é usada para o efeito. Imaginemos que estamos na presença de dois *qubits*, interessando-nos "trocar" o valor do *qubit alvo* se o *qubit de controlo* estiver a 0. A Figura 3.6 apresenta a sintaxe correta para o efeito assim como uma possível equivalência que traduz a mesma ação.

Até ao momento vimos um exemplo simplista do que são *operações controladas*. Na realidade o número de *qubits* usados nas aplicações quânticas não é tão reduzido. Daí a necessidade de existirem operadores, capazes de manusearem vários *qubits de controlo*, assim como *qubits alvo* ao mesmo tempo. Seja $n+k$ o número de *qubits* existentes, **U** um operador que atue em k *qubits*. Então a *operação controlada* C^n(U), é dada por:

$$C^n(U)|x_1x_2\ldots x_n\rangle|\psi\rangle = |x_1x_2\ldots x_n\rangle U^{x_1x_2\ldots x_n}|\psi\rangle \tag{3.46}$$

Desta forma, ao exponenciar U a $x_1x_2\ldots x_n$, garantimos que o operador é apenas aplicado aos últimos k *qubits* do sistema e apenas na condição de o expoente ou melhor, o *produto* dos *bits* do expoente ser igual a 1. Se definirmos um operador V unitário tal que, $V^2 = U$ podemos afirmar que a operação unitária $C^2(U)$, pode ser implementada de acordo com a Figura 3.7.

Como exemplo, podemos implementar o operador *Toffoli*. Considerando a Figura 3.8, onde temos expressa a sua tabela de verdade, bem como a sua representação num circuito clássico. Deduzimos tratar-se da operação $C^2(X)$. Precisamos agora de definir V tal que, $V^2 = X$. Pelo Teorema da Decomposição Espectral (Teorema 1), sabemos que,

$$X = \sum_i \lambda_i |v_i\rangle\langle v_i|, \tag{3.47}$$

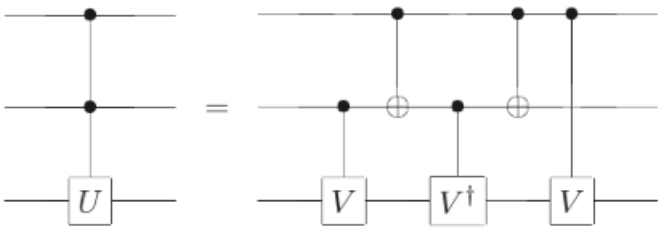


Figura 3.7: Circuito para que implementa uma operação do tipo $C^2(U)$. V é um operador unitário, tal que, $V^2 = U$

Inputs			Outputs		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

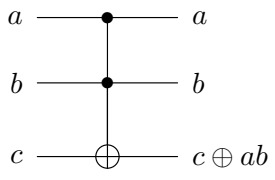


Figura 3.8: Tabela de Verdade de Operador *Toffoli* bem como o seu Circuito clássico

sendo λ_i e v_i respectivamente os valores e vectores próprios de X. Estes por sua vez são,

$$v_1 = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \text{com } \lambda_1 = 1; \tag{3.48}$$

$$v_2 = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \text{com } \lambda_2 = -1, \tag{3.49}$$

e apresentam-se normalizados. Assim, $V = \sqrt{X}$ é dada por,

$$V = 1 \times |v_1\rangle\langle v_1| + i \times |v_2\rangle\langle v_2|, \tag{3.50}$$

o que nos permite obter,

$$V = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}. \tag{3.51}$$

Recorrendo à Figura 3.8 podemos também construir a representação matricial do operador *Toffoli*. É mais uma vez trivial perceber o que acontece a cada um dos possíveis estados, apenas observando as colunas deste operador.

$$\text{Toffoli} \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.52}$$

Apesar de termos introduzido a notação para a implementação de um unitário U, com um número arbitrário de *qubits* de controlo, não explicamos como concretizavamos a mesma num circuito quântico. Optou-se por apresentar primeiro o operador *Toffoli*, uma vez que este é fundamental para o resultado pretendido. Considere-se a Figura 3.9, que apresenta o circuito que implementa um qualquer unitário para um sistema de n = 5 *qubits* de controlo. Pela análise da mesma, percebemos que existem três diferentes tipos de *qubits*. Os mesmos estão devidos em, *qubits* de controlo, com a mesma

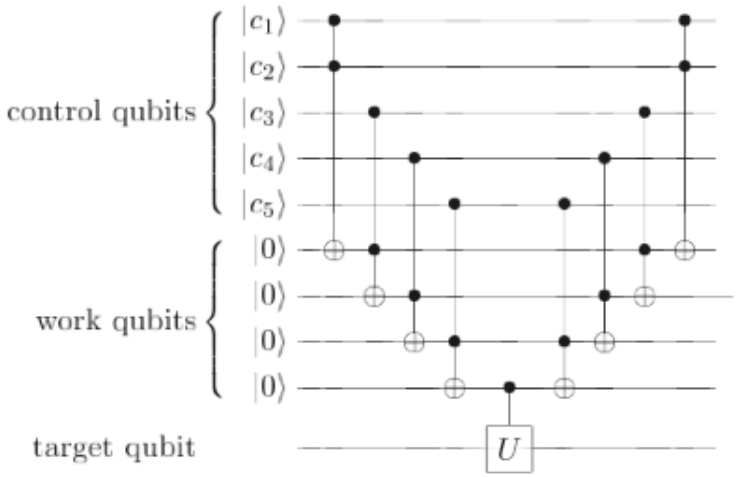


Figura 3.9: Circuito para que implementa uma operação do tipo $C^n(U)$. No exemplo n = 5

utilidade já descrita, o *qubit* alvo, cuja funcionalidade é obvia e os chamados *qubits* de trabalho. Vamos concentrar-nos nestes últimos. Resumidamente, eles vão guardar o estado da "análise" aos *qubits* de controlo. Analisando o circuito observamos que é feito o *AND* entre os dois primeiros *qubits* de controlo, c_1 e c_2, através da aplicação do operador *Toffoli*. O resultado desta operação é guardado no primeiro *qubit* de trabalho. Por seu lado este valor é usado para um novo *AND* com o *qubit* de controlo seguinte, recorrendo novamente ao operador *Toffoli*, sendo o resultado guardado no segundo *qubit* de trabalho. Continuando assim recursivamente. É fácil de inferir que precisamos de (n-1) *qubits* de trabalho para um sistema de n *qubits* de controlo. Neste momento é então efetuada a operação controlada $\mathbb{U}$. Por último, as alterações ao nível dos *qubits* de trabalho são desfeitas, voltando os mesmo ao estado inicial $|0\rangle$.

De seguida avançaremos para a última seção deste capítulo. Nela procuraremos demonstrar que existe um conjunto reduzido de operadores quânticos, que proporcionam universalidade. Por outras palavras é possível recriar uma boa aproximação a qualquer circuito clássico ou quântico, apenas com um conjunto pequeno de operadores que iremos apresentar.

3.4 Operadores Quânticos Universais

Nesta seção vamos abordar o problema da universalidade dos operadores. Para tal, começamos por explicar que todos os circuitos clássicos podem ser implementados por um circuito quântico equivalente, obtendo desta forma universalidade do ponto de vista clássico. Em seguida apresentaremos dois algoritmos que nos permitirão concluir sobre a existência de um conjunto de operadores que provam universalidade, mas agora do ponto de vista quântico. Da junção destas duas demonstrações, advém um dos pontos fortes da computação quântica: a possibilidade de substituir o modelo de computação clássico, pelo menos em teoria.

3.4.1 Um Conjunto Clássico Universal

Da computação clássica sabemos que o conjunto **AND**, **OR**, **NOT** é universal, ou seja é suficiente para construir todos os outros operadores lógicos. Vamos analisar de seguida alguns operadores clássicos, incluindo os citados para melhor compreendermos a questão da universalidade. Começamos pelo operador **Not** cujo comportamento é bastante trivial trocando apenas o *bit* de *input*:

Inputs	Outputs
0	1
1	0

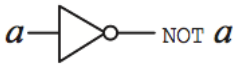


Figura 3.10: Tabela de verdade de operador *NOT* bem como o seu circuito clássico

O operador **AND** produz o *output* 1 apenas se os dois bits de *input* tiverem o valor 1.

Inputs		Outputs
0	0	0
0	1	0
1	0	0
1	1	1

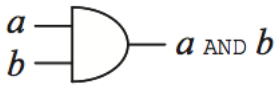


Figura 3.11: Tabela de verdade de operador *And* bem como o seu circuito clássico

O operador **OR** ao invés, devolve o valor 1, sempre que um dos dois *bits* de *input* tenha esse valor.

Inputs		Outputs
0	0	0
0	1	1
1	0	1
1	1	1

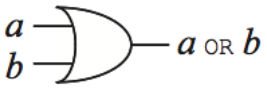


Figura 3.12: Tabela de verdade de operador *OR* bem como o seu circuito clássico.

O operador **XOR** já introduzido, produz o valor 1 se e somente se apenas um dos *bits* de input tiver o valor 1.

Inputs		Outputs
0	0	0
0	1	1
1	0	1
1	1	0

Figura 3.13: Tabela de verdade de operador *XOR* bem como o seu circuito clássico.

O operador **NAND** produz o *output* 0 se e só se, os dois *inputs* tiverem o valor 1, retornando 1 nos outros casos. Isto é conseguido, aplicando o operador **NOT** ao *output* do operador **AND**. Em suma:

Inputs		Outputs
0	0	1
0	1	1
1	0	1
1	1	0

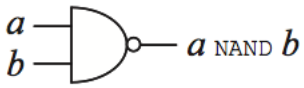


Figura 3.14: Tabela de verdade de operador *Nand* bem como o seu circuito clássico.

Este último operador é de tal forma especial, que permite construir todos os outros (**NOT**,**AND**, **XOR**, **OR**). Resumidamente:

Inputs	Outputs
0	1
1	0

Figura 3.15: Tabela de verdade de operador *NOT* bem como o seu circuito clássico implementado usando o operador NAND.

O operador **AND** é implementado aplicando o operador **NAND**, seguido do operador **NOT**. Isto devolve um **NOTAND** isto é **AND**:

Inputs		Outputs
0	0	0
0	1	0
1	0	0
1	1	1

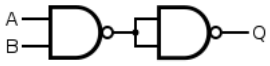


Figura 3.16: Tabela de verdade de operador *AND* bem como o seu circuito clássico implementado usando o operador NAND

Pela tabela de verdade do operador **OR**, sabemos que basta que um dos *inputs* seja 1 para o *output* ser 1. Para além disso, também sabemos que quando no operador **NAND** obtemos o valor 0 no *output* é porque os dois *inputs* são 1. A solução passa então por inverter o *input*, "forçando" em caso de *input* positivo o *output* positivo. Ou seja:

Inputs		Outputs
0	0	0
0	1	1
1	0	1
1	1	1

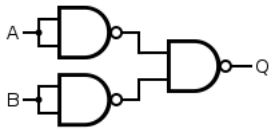


Figura 3.17: Tabela de verdade de operador *OR* bem como o seu circuito clássico implementado usando o operador NAND.

Por último, o operador **XOR** é conseguido utilizando o mesmo esquema do **OR**, mas adicionando mais um **NAND**, garantido que quando o *input* é 1 nos dois *bits*, o *input* no último **NAND** também o seja, o que permite que o *output final* seja 0.

Inputs		Outputs
0	0	0
0	1	1
1	0	1
1	1	0

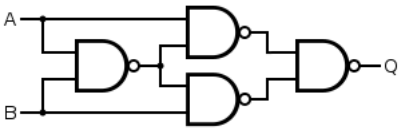


Figura 3.18: Tabela de verdade de operador *XOR* bem como o seu circuito clássico implementado usando o operador NAND.

Neste momento, já conseguimos perceber que o operador **NAND** consegue substituir o operador **AND**,**OR**,**XOR** e **NOT**. Por outro lado também já percebemos que tanto uma opção como a outra acarreta o problema da irreversibilidade[4], à exceção do operador clássico **NOT** que é reversível. A resposta a este dois problemas, isto é, universalidade e facto de precisarmos de computação reversível, surge do operador **TOFFOLI** por nós já analisado nas duas vertentes (clássica e quântica). Vamos por agora apenas considerar a sua implementação clássica. Percebemos que esta oferece reversibilidade uma vez que os dois *bits* na entrada aparecem inalterados na saída. Para além disso, garante-nos universalidade, uma vez que o operador **TOFFOLI** contém o operador **NAND** no seu "interior". Quando o terceiro bit tem o valor 1, o operador **TOFFOLI** escreve o **NAND** dos primeiros dois *bits* no terceiro *bit*. Depois de tudo isto, conseguimos finalmente concluir que o operador **TOFFOLI** é por si universal e reversível. Este resultado é de suma importância, dado que a nível quântico também possuímos este operador. Tal permite afirmar que, existe **um conjunto particular de operadores quânticos, universal do ponto de vista clássico**.

Vamos agora passar para a vertente quântica, identificando um conjunto de operadores universais da mesma.

[4]depois de aplicado o operador o input é destruido

3.4.2 Unitários de nível 2.

Os *unitários de nivel 2* é o nosso ponto de partida na procura por um conjunto universal de operadores quânticos. Este algoritmo considera um operador **U** que atua num sistema de *n qubits*. Como já analisamos em **Fundamentos da Álgebra Linear**, este operador opera num espaço vectorial denominado de *Hilbert*, cuja dimensão é $d = 2^n$. Isto vai implicar que o operador terá uma representação matricial de dimensão $d \times d$. A exposição do algoritmo que se segue é adaptada de Nielsen and Chuang [2000][Pag 189].

O mote deste algoritmo é então definir um conjunto de matrizes $U_{d-1} \dots U_1$, que atuem por si em apenas dois ou menos componentes desse espaço vectorial, tal que :

$$U_{d-1}U_{d-2} \dots U_1 = I \tag{3.53}$$

assim como:

$$U = U_1^\dagger \dots U_{d-2}^\dagger U_{d-1}^\dagger \tag{3.54}$$

Ao invés de apenas mencionarmos o algoritmo, vamos acompanhá-lo de um exemplo prático, para facilitar a sua compreensão. É importante salientar que o mesmo foi implementado recorrendo à plataforma **SAGE** podendo o código ser consultado no anexo D.1. Considere-se então o seguinte operador F, que corresponde a um caso especial da *transformada de Fourier* quântica que estudaremos no próximo capítulo.

$$F \equiv \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}. \tag{3.55}$$

Pretendemos encontrar as matrizes, tais que:

$$k \leq \frac{d(d-1)}{2}. \tag{3.56}$$

O que no nosso exemplo se traduz em descobrir, $V_1, V_2, V_3, V_4, V_5, V_6$ para o operador **F**. Genericamente, seja V um operador unitário (o nosso **F**),

$$V \equiv \begin{bmatrix} a & e & j & n \\ b & f & k & o \\ c & g & l & p \\ d & h & m & q \end{bmatrix} \tag{3.57}$$

Usamos o procedimento seguinte para construir V_1: se $b = 0$ então:

$$V_1 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.58}$$

Se por outro lado $b \neq 0$ então:

$$V_1 \equiv \begin{bmatrix} \frac{a^*}{\sqrt{\|a\|^2+\|b\|^2}} & \frac{b^*}{\sqrt{\|a\|^2+\|b\|^2}} & 0 & 0 \\ \frac{b}{\sqrt{\|a\|^2+\|b\|^2}} & \frac{-a}{\sqrt{\|a\|^2+\|b\|^2}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.59}$$

No nosso exemplo, a aplicação deste procedimento resulta em:

$$V_1 \equiv \begin{bmatrix} \frac{0.5^*}{\sqrt{0.5^2+0.5^2}} & 0.7071 & 0 & 0 \\ 0.7071 & -0.7071 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.60}$$

Atualmente se multiplicarmos $V_1 \times F$, reparamos que o elemento da segunda linha, primeira coluna, passou para 0.

$$V_1 V \equiv \begin{bmatrix} a' & e' & j' & n' \\ 0 & f' & k' & o' \\ c' & g' & l' & p' \\ d' & h' & m' & q' \end{bmatrix}. \equiv \begin{bmatrix} 0.7071 & 0.3536+0.3536i & 0 & 0.3536i \\ 0 & 0.3536-0.3536i & 0.7071 & 0.3536+0.3536i \\ 0.5 & -0.5 & 0.5 & -0.5 \\ 0.5 & 0.5i & -0.5 & 0.5 \end{bmatrix} \tag{3.61}$$

O resto do algoritmo segue a mesma ideia, anulando cada um dos restantes elementos da primeira coluna, antes de passar para a segunda e por aí adiante, anulando todos os elementos abaixo da diagonal. Voltando a V_2, queremos anular $\mathbf{c}$, caso ele seja $\neq 0$(caso contrário têm-se $V_2 = I$). Para isso precisamos de uma matriz com a seguinte configuração:

$$V_2 \equiv \begin{bmatrix} \frac{a'^*}{\sqrt{\|a'\|^2+\|c'\|^2}} & 0 & \frac{c'^*}{\sqrt{\|a'\|^2+\|c'\|^2}} & 0 \\ 0 & 1 & 0 & 1 \\ \frac{c'}{\sqrt{\|a'\|^2+\|c'\|^2}} & 0 & \frac{-a'}{\sqrt{\|a'\|^2+\|c'\|^2}} & \\ 0 & 0 & 0 & 1 \end{bmatrix}. \equiv \begin{bmatrix} \frac{0.7071}{\sqrt{\|0.7071\|^2+\|0.5\|^2}} & 0 & \frac{0.5}{\sqrt{\|0.7071\|^2+0.5^2}} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{0.5}{\sqrt{\|0.7071\|^2+\|0.5\|^2}} & 0 & \frac{-0.7071}{\sqrt{\|0.7071\|^2+\|0.5\|^2}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.62}$$

V_2 fica então a seguinte matriz:

$$V_2 = \begin{bmatrix} 0.8165 & 0 & 0.5774 & 0 \\ 0 & 1 & 0 & 0 \\ 0.5774 & 0 & -0.8165 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.63}$$

Ao verificarmos se $\mathbf{c} = 0$, isto é $V_2 V_1 V$, confirma-mos que correu tudo como esperado:

$$V_2 V_1 V \equiv \begin{bmatrix} 0.8165 & 0.2887i & 0.2887 & -02887i \\ 0 & 0.3536-0.3536i & 0.7071 & 0.3536+0.3536i \\ 0 & 0.6124+0.2041i & -0.4083 & 0.6124-0.204i \\ 0.5 & 0.5i & -0.5 & 0.5i \end{bmatrix}. \tag{3.64}$$

Falta agora anular o último elemento desta coluna. Para tal, usaremos os valores resultantes de $V_2 V_1 V$, marcando esses valores usando $''$. Posto isto,

V_3 é calculado da seguinte forma:

$$\begin{bmatrix} \frac{a''^*}{\sqrt{\|a''\|^2+\|d''\|^2}} & 0 & 0 & \frac{d''^*}{\sqrt{\|a''\|^2+\|d''\|^2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{d''}{\sqrt{\|a''\|^2+\|d''\|^2}} & 0 & 0 & \frac{-a''}{\sqrt{\|a'\|^2+\|d''\|^2}} \end{bmatrix} \equiv \begin{bmatrix} \frac{0.866}{\|0.866\|^2+\|0.5\|^2} & 0 & 0 & \frac{0.5}{\sqrt{\|0.866\|^2+\|0.5\|^2}} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \frac{0.5}{\sqrt{\|0.866\|^2+\|0.5\|^2}} & 0 & 0 & \frac{-0.866}{\sqrt{\|0.866\|^2+\|0.5\|^2}} \end{bmatrix}. \tag{3.65}$$

Finalmente, obtemos o resultado desejado, com todos os elementos da coluna 1 abaixo do elemento da diagonal, iguais a zero. Isto é, $V_3V_2V_1V$ em que

$$V_3 = \begin{bmatrix} 0.866 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0.5 & 0 & 0 & -0.866 \end{bmatrix}. \tag{3.66}$$

As restantes colunas vão seguir o mesmo *modus operandi*, motivo pelo qual apenas apresentaremos as matrizes resultantes, ou seja, $V_4 e V_5$ resultantes da segunda coluna,

$$V_4 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.433+0.433i & 0.75-0.25i & 0 \\ 0 & 0.75+0.25i & 0.433+0.433i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \; V_5 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.8165 & 0 & -0.5774i \\ 0 & 0 & 1 & 0 \\ 0 & 0.5774i & 0 & -0.8165 \end{bmatrix} \tag{3.67}$$

Ou seja, neste momento temos a seguinte matriz, dada por $V_5V_4V_3V_2V_1V$,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7071 & 0.7071i \\ 0 & 0 & -0.7071 & 0.7071i \end{bmatrix}. \tag{3.68}$$

A última decomposição (V_6), consiste em efetuar a seguinte operação,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e'' & h'' \\ 0 & 0 & f'' & j'' \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e''^* & f''^* \\ 0 & 0 & h''^* & j''^* \end{bmatrix}. \tag{3.69}$$

V_6 é como tal facilmente calculada,

$$V_6 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.7071 & -0.7071 \\ 0 & 0 & -0.7071i & -0.7071i \end{bmatrix}. \tag{3.70}$$

Ao verificarmos os nosso resultados, nas condições inicialmente apresentadas (3.53) e (3.54), obtemos a confirmação de como conseguimos decompor um operador unitário num produto de simples *two-level unitaries matrices*. Este algoritmo como seria esperado, escala para operadores de maiores dimensões e juntamente com a próxima seção vai permitir-nos concluir sobre a universalidade de alguns operadores quânticos.

3.4.3 Códigos de Gray

Na seção anterior, observamos como um operador unitário pode ser decomposto num conjunto de matrizes *unitárias de nível 2*. Apesar de a abordagem escolhida para analisar um exemplo concreto, o algoritmo está desenhado para decompor um qualquer operador unitário. Nesta seção analisaremos outro algoritmo, cuja função é por mostrar que, qualquer operação *two level unitary* pode ser implementada recorrendo a um único *qubit* e operadores **CNOT**. Ao combinar este resultado com o demonstrado na seção anterior, provamos que operações num único *qubit* e operadores *CNOT* formam um conjunto universal para computação quântica. Na exposição do presente algoritmo, seguimos de perto a abordagem apresentada em Nielsen and Chuang [2000][Pag. 191].

Vamos supor que U é um operador unitário de nível 2 que atua num sistema de n *qubits*. Mais ainda, o comportamento de U é não trivial no espaço vectorial descrito pela base computacional $|s\rangle$ e $|t\rangle$. Analisando as bases computacionais pela sua expansão binária, isto é, $s = s_1 \dots s_n$ e $t = t_1 \dots t_n$, podemos afirmar que existe uma *sub-matriz* de U, dada por $\tilde{U}$, que descreve uma operação unitária em apenas 1 *qubit*. Vamos concentrar todos estes conceitos num exemplo concreto. Considere-se o seguinte operador U definido na seguinte base computacional $|000\rangle \dots |111\rangle$.

$$U \equiv \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 & 0 & c \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ b & 0 & 0 & 0 & 0 & 0 & 0 & d \end{bmatrix} \tag{3.71}$$

tal que, *a,b,c,d* são números complexos e $\tilde{U}$ é a *sub-matriz* unitária:

$$\tilde{U} \equiv \begin{bmatrix} a & c \\ b & d \end{bmatrix} \tag{3.72}$$

Ao analisarmos a base computacional, verificamos que U atua não trivialmente nos estados $|000\rangle$ e $|111\rangle$. A ideia do algoritmo dos *códigos Gray* passa por, começar por escrever um código binário que ligue esses estados tal que, o código comece por 000 e termine em 111 podendo os elementos adjacentes ao mesmo, deferir num elemento apenas, ou seja

$$|g_1\rangle = 000, \tag{3.73}$$
$$|g_2\rangle = 001, \tag{3.74}$$
$$|g_3\rangle = 011, \tag{3.75}$$
$$|g_4\rangle = 111. \tag{3.76}$$
$$\tag{3.77}$$

Considerando os dois primeiros estados $|g_1\rangle$ e $|g_2\rangle$, sabemos que estes diferem num determinado digito i. O procedimento a seguir passa então por realizar uma operação controlada, condicionada nos *bits* idênticos entre $|g_1\rangle$ e $|g_2\rangle$ e que troque o valor do bit (**CNOT**) no bit divergente. Isto é então conseguido através da aplicação sucessiva de operadores *CNOT*, que vão efetuando as trocas $|g_1\rangle \rightarrow |g_2\rangle \ldots |g_{t-1}\rangle$. Neste momento aplicamos o operador $\tilde{U}$ controlado, sendo o *qubit alvo* localizado no elemento(isto é, *bit*) onde $|g_{t-1}\rangle$ difere de $|g_t\rangle$. A fase final do algoritmo desfaz todas as operações controladas realizadas inicialmente (ate à aplicação do operador $\tilde{U}$).
Este comportamento pode ser comprovado pelo circuito que em seguida se apresenta, descrevendo o exemplo em cima apresentado:

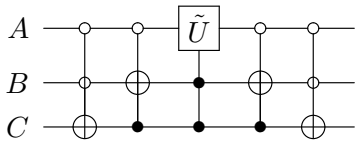


Figura 3.19: Circuio que implementa a operação (3.71)

É fácil de comprovar que todos os restantes estados atravessam este circuito sem serem alterados. Podemos afirmar que o circuito da Figura 3.19 reproduz a operação descrita por U. Estas duas últimas seções permitem-nos concluir que *CNOT* e unitários que atuam em *qubits* simples formam um conjunto universal para a computação quântica. No entanto, e dado que o conjunto de operações unitárias sobre um *qubit* é infinito, este resultado não nos permite identificar um conjunto restrito de operadores quânticos que seja universal para a computação quântica. Na próxima seção veremos que esse conjunto de facto existe, se relaxarmos o requisito para se obter um circuito que se comporta como uma aproximação (tão boa quanto possível) do circuito original.

3.4.4 Operadores H + S + CNOT + T − O Conjunto Universal

Esta seção encerra um dos pontos chave sobre a computação quântica, a *universalidade*. Nas próximas linhas, procuramos apresentar a prova que o conjunto *H,T,S, CNOT* é capaz de proporcionar uma boa aproximação a qualquer operador unitário quântico. Para o fazermos, vamos recorrer a obra de Ouellette [2002][pag. 29], transcrevendo desta a prova que demonstra exatamente o que pretendemos.

Vamos considerar o operador T(*fase global*) e a combinação *HTH*. Relembrando o que foi explicado em *Operações num qubit*, mais precisamente as equações (3.20),(3.21),(3.22), podemos afirmar que o operador T se traduz

numa rotação $\frac{\pi}{4}$ sobre $\hat{z}$, mais ou menos uma diferença de fase:

$$T \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \equiv R_z(\frac{\pi}{4}) \equiv \begin{bmatrix} e^{-\frac{i\pi}{8}} & 0 \\ 0 & e^{-\frac{i\pi}{8}} \end{bmatrix} \equiv e^{\frac{i\pi}{8}} \begin{bmatrix} e^{-\frac{i\pi}{8}} & 0 \\ 0 & e^{-\frac{i\pi}{8}} \end{bmatrix} \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \tag{3.78}$$

De maneira semelhante, a combinação HTH pode ser vista como, uma rotação em torno de $\hat{x}$ segundo um angulo de $\frac{\pi}{4}$ radianos, mais ou menos uma diferença de fase.

$$\begin{aligned} HTH \equiv & \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ \equiv & \frac{1}{2} \begin{bmatrix} (1+e^{\frac{i\pi}{4}}) & (1-e^{\frac{i\pi}{4}}) \\ (1-e^{\frac{i\pi}{4}}) & (1+e^{\frac{i\pi}{4}}) \end{bmatrix} \end{aligned} \tag{3.79}$$

já que,

$$\begin{aligned} 1+e^{\frac{i\pi}{4}} =& 1+\cos\frac{\pi}{4}+i\sin\frac{\pi}{4} = 2\cos^2\frac{\pi}{8}+2i\sin\frac{\pi}{8}\cos\frac{\pi}{8} \\ =& 2e^{\frac{i\pi}{8}}\cos\frac{\pi}{8} \end{aligned} \tag{3.80}$$

$$\begin{aligned} 1-e^{\frac{i\pi}{4}} =& 1-\cos\frac{\pi}{4}-i\sin\frac{\pi}{4} = 2\sin^2\frac{\pi}{8}-2i\sin\frac{\pi}{8}\cos\frac{\pi}{8} \\ =& -2e^{\frac{i\pi}{8}}\sin\frac{\pi}{8} \end{aligned} \tag{3.81}$$

o que implica que,

$$HTH = e^{\frac{i\pi}{8}} \begin{bmatrix} \cos\frac{\pi}{8} & -i\sin\frac{\pi}{8} \\ -i\sin\frac{\pi}{8} & \cos\frac{\pi}{8} \end{bmatrix} \equiv R_x\left(\frac{\pi}{4}\right) \tag{3.82}$$

Ao combinarmos as duas rotações dos dois operadores, obtemos

$$\begin{aligned} e^{\frac{-i\pi Z}{8}} e^{\frac{-i\pi X}{8}} \equiv & (\cos\frac{\pi}{8} - iZ\sin\frac{\pi}{8})(\cos\frac{\pi}{8} - iX\sin\frac{\pi}{8}) \\ \equiv & \cos^2\frac{\pi}{8} - i\left[(X+Z)\cos\frac{\pi}{8} + Y\sin\frac{\pi}{8}\right]\sin\frac{\pi}{8} \end{aligned} \tag{3.83}$$

Ao comparamos esta equação com a Equação 3.19 definida na seção *Operações num qubit*, verificamos que temos uma rotação em torno de um vector/eixo n expressa por, $(\cos\frac{\pi}{8}, \sin\frac{\pi}{8}, \cos\frac{\pi}{8})$, com um angulo ϕ dado por $\cos\frac{\phi}{2} = \cos^2\frac{\pi}{8}$. Apesar de não se provar, este ϕ é um número irracional múltiplo de 2π[5]. Então é possível para um qualquer θ, aplicar sucessivas rotações, $R_n(\phi)$ capazes de aproximar uma rotação por qualquer ângulo, tal que

$$e^{ik\pi\lambda} \simeq e^{i\theta}, \text{onde } k \text{ é um qualquer número inteiro.} \tag{3.84}$$

[5] $\phi = 2\lambda\pi$, onde λ é um irracional

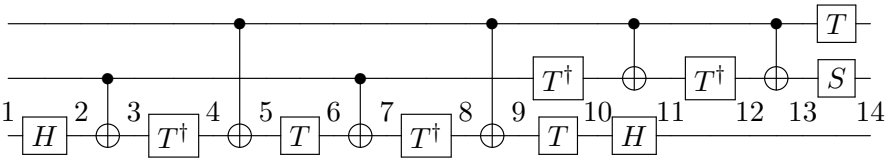


Figura 3.20: Implementação do operador Toffoli usando apenas os operadores, CNOT,$\frac{\pi}{8}$, Hadamard, Fase

uma vez que, os fatores da *fase* são definidos modulo 2π. Vamos agora finalmente analisar o produto de $HR_n(\phi)H$, considerando o facto de H = $\frac{1}{\sqrt{2}}(X+Z)$, sendo H unitária e*Hermitiana*.

$$HR_n(\phi)H = \cos^2\frac{\pi}{8} - i\left[(X+Z)\cos\frac{\pi}{8} + HYH\sin\frac{\pi}{8}\right]\sin\frac{\pi}{8}. \tag{3.85}$$

uma vez que,

$$\begin{aligned} HYH =& \frac{1}{2}(X+Z)Y(X+Z) = \frac{1}{2}(X+Z)(-iZ+iX) \\ =& \frac{1}{2}(-Y-Y) = -Y. \end{aligned} \tag{3.86}$$

temos,

$$HR_n(\theta)H = \cos^2\frac{\pi}{8} - i\left[(X+Z)\cos\frac{\pi}{8} - Y\sin\frac{\pi}{8}\right]\sin\frac{\pi}{8}. \tag{3.87}$$

O que se traduz numa rotação no eixo $m = (\cos\frac{\pi}{8}, -\sin\frac{\pi}{8}, \cos\frac{\pi}{8})$, assim sendo, verificamos que m e n não são paralelos, o que nos garante que podemos representar qualquer *operador num único qubit* como um produto de rotações $R_n(\theta)$ e $R_m(\theta)$, mais ou menos uma diferença de fase.

Dado que estas rotações foram implementadas recorrendo aos operadores T e H, se a isto juntarmos o *CNOT*, proveniente dos resultados/algoritmos obtidos nos *unitários de nível 2* e *códigos de Gray*, provamos que o **conjunto H,T,CNOT é universal**. Como exemplo da aplicação deste conjunto, considere-se a Figura 3.20 onde se apresenta uma possível implementação do operador *Toffoli* retirada de Nielsen and Chuang [2000]. Por forma a corroborar esse propósito, vamos fazer uma análise da mesma. Em primeiro lugar, vamos considerar a ação dos operadores presentes pela ordem apresentada, isto é seguindo a ordem numérica. Através das identidades expressas em (3.38), verificamos que:

$$XZX = -Z \longrightarrow XT^\dagger X = e^{-i\frac{\pi}{4}}T \tag{3.88}$$

Este raciocínio permite-nos saltar diretamente para o passo 9 do circuito.

Assim sendo, temos:

$$|\psi_1\rangle = |x, y, z\rangle \tag{3.89}$$
$$|\psi_9\rangle = |x, y\rangle \otimes X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.90}$$
$$|\psi_{10}\rangle = |x\rangle \otimes T^\dagger|y\rangle \otimes T X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.91}$$
$$|\psi_{11}\rangle = |x\rangle \otimes X^x T^\dagger|y\rangle \otimes H T X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.92}$$
$$|\psi_{12}\rangle = |x\rangle \otimes T^\dagger X^x T^\dagger|y\rangle \otimes H T X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.93}$$
$$|\psi_{13}\rangle = |x\rangle \otimes X^x T^\dagger X^x T^\dagger|y\rangle \otimes H T X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.94}$$
$$|\psi_{14}\rangle = e^{ix\frac{\pi}{4}} \otimes S X^x T^\dagger X^x T^\dagger|y\rangle \otimes H T X^x T^\dagger X^y T X^x T^\dagger X^y H|z\rangle \tag{3.95}$$

Em $|\psi_{14}\rangle$ note-se como a aplicação do operador T ao *qubit* x, resultou em $e^{ix\frac{\pi}{4}}|x\rangle$. Sendo assim e fazendo $x = 0$, verificamos que:

$$|\psi_{out}\rangle = |0\rangle \otimes |y\rangle \otimes |z\rangle = TOFFOLI|0, y, z\rangle \tag{3.96}$$

Se por outro lado, $x = 1$, obtemos:

$$e^{i\pi\frac{x}{4}} S X^x T^\dagger X^x T^\dagger|y\rangle = e^{i\frac{\pi}{4}} S X T^\dagger X T^\dagger|y\rangle \tag{3.97}$$

Neste momento podemos usar a "identidade" 3.88, ficando

$$e^{i\frac{\pi}{4}} \times e^{-i\frac{\pi}{4}} \times S \times T \times T^\dagger|y\rangle \equiv 1 \times S \times I|y\rangle = i^y|y\rangle \tag{3.98}$$

de seguida, fazendo $y = 0$, obtemos

$$|\psi_{out}\rangle = |1\rangle \otimes |0\rangle \otimes H T X T^\dagger T X T^\dagger H|z\rangle = |1, 0, z\rangle \tag{3.99}$$

uma vez que, $(TXT^\dagger)^2 \equiv H^2 = I$. Por ultimo, fazendo $x = y = 1$, obtemos:

$$|\psi_{out}\rangle = |1, 1\rangle \otimes iH T X T^\dagger X T X T^\dagger X H|z\rangle \tag{3.100}$$

Utilizando a "identidade" 3.88, sabemos que:

$$X T^\dagger X = \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & 1 \end{bmatrix} \times T \Rightarrow (TXT^\dagger X)^2 = -iZ. \tag{3.101}$$

então é trivial concluir por 3.38 que:

$$|\psi_{out}\rangle = |1, 1\rangle \otimes HZH|z\rangle = |1, 1\rangle \otimes X|z\rangle = TOFFOLI|1, 1, z\rangle. \tag{3.102}$$

3.5 Sumário

Este capítulo centrou-se entre dois objetivos claros. Primeiro, procurou transmitir em detalhe o funcionamento lógico por detrás deste novo modelo de computação, realizável através de *circuitos quânticos*. Em segundo lugar, explicou que existe um conjunto muito reduzido de operadores lógicos capazes de oferecer universalidade, tanto do ponto de vista clássico, como quântico. Além disso, permitiu-nos um primeiro contato com o tema do próximo Capítulo, *Algoritmos Quânticos*.

No próximo Capítulo, focaremos então atenções em determinados algoritmos quânticos. Para tal, abordaremos um dos seus elementos chave - *parelismo quântico*, de onde advém a enorme capacidade de processamento que os mesmos exploram e que nos incentiva a procurar um novo esquema ou modelo criptográfico.

Capítulo 4

Algoritmos Quânticos

The theory of computation has traditionally been studied almost entirely in the abstract, as a topic in pure mathematics. This is to miss the point of it. Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone, and not by pure mathematics.

David Deutsch

Até ao momento, a nossa discussão teve como mote a apresentação do modelo de computação quântico. Procurou-se demonstrar que este modelo, conseguia simular o modelo clássico e determinista atual. No seguimento desta ideia facilmente surge a questão: *"E se o sistema clássico que pretendíamos simular fosse não deterministico?"* ou seja, tivesse a capacidade de gerar *bits* aleatórios para realizar computação. Relembrando o que aprendemos sobre a medição de um estado quântico, podemos facilmente demonstrar que o modelo quântico consegue igualar esta propriedade. Por exemplo, se sujeitarmos o estado $|0\rangle$ ao operador H, obtemos o estado,

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}. \tag{4.1}$$

Sabemos que após medição do mesmo temos 50% de hipóteses de este gerar $|0\rangle$ ou $|1\rangle$ ou seja, esta propriedade dota o modelo quântico de capacidade não determinista. No entanto e apesar de este ser um resultado importantíssimo, não é propriamente a capacidade de o modelo quântico emular o modelo clássico que nos motiva. O nosso interesse advém da enorme capacidade de processamento proveniente do uso de *qubits* e *operadores quânticos*. Este capítulo serve o propósito de explicar essa mesma capacidade. Para o efeito, apresentam-se versões simplificadas dos algoritmos inicialmente propostos por *Deutsch* e *Deutsch e Jozsa*, retiradas de Nielsen and Chuang [2000] e Ouellette [2002]. Além destes algoritmos, discorreremos também sobre o

algoritmo da transformada de *Fourier*. Esta travessia baseia-se no trabalho de Cheung [2003]. Com ela procuramos explicar a sub-rotina da transformada de *Fourier* que se encontra no cerne do *Algoritmo de Shor*, que será abordado no capítulo seguinte.

4.1 O ingrediente secreto - Paralelismo Quântico

O conceito de *paralelismo quântico* é responsável por acelerar determinadas tarefas em computadores quânticos, que se pensa serem impossíveis de igualar, em processamento e memória, num computador clássico. De um ponto de vista simplista, podemos dizer que esta funcionalidade permite *calcular* uma determinada função $f(x)$, para vários x ao mesmo tempo. Fazendo a ponte com computação quântica, podemos preparar um *registo* em superposição de n *qubits* tal que, o mesmo consegue representar todos os estados base isto é, todos os números entre 0 e $2^n - 1$ (diversos x). De seguida se sujeitarmos esse *registo* a um operador unitário (a nossa função f), este vai alterar todos os estados base definidos. Nesta construção, o primeiro registo é conhecido por *data* e o segundo por *target*. Vamos analisar um exemplo que concentre todos estes conceitos. Considere-se a seguinte função,

$$f(x) : \{0,1\} \rightarrow \{0,1\}. \tag{4.2}$$

Uma forma de computar esta função num computador quântico é definir o *registo*, $|x, y\rangle$ de dois *qubits*. De seguida, utilizando o conjunto apropriado de operadores é possível evoluir o estado inicial para, $|x, y \oplus f(x)\rangle$, onde $\oplus$ significa a adição módulo 2. Preparando corretamente o estado de entrada, podemos computar a dada função para os valores 0 e 1. Considere-se o seguinte circuito,

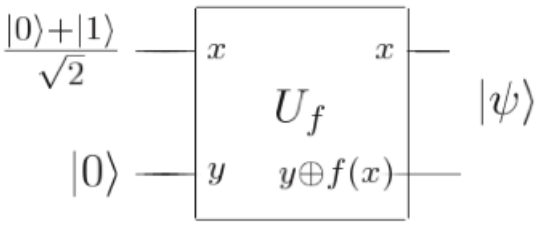


Figura 4.1: Circuito quântico que avalia *simultaneamente* f(0) e f(1). À transformação dada pela aplicação $|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$, deu-se o nome de U_f.

sendo o estado $|\psi\rangle$, dado por,

$$\frac{|0, f(0)\rangle + |1, f(1)\rangle}{\sqrt{2}}. \tag{4.3}$$

Analisando-o com atenção, vemos que estamos na presença de um estado quântico "fenomenal". A sobreposição, possui informação sobre de f(0) e f(1).

É importante salientar que, a mesma noção de paralelismo num computador clássico, só seria atingida duplicando o circuito, computando distintamente f(0) e f(1) em cada porção. No entanto, relembrando o que já sabemos sobre um estado com estas características concluímos que, ao acedermos ao estado ele vai *colapasar* deixando-nos apenas com $f(0)$ ou $(f1)$. Veremos nas próximas páginas que apesar de isto ser verdade é possível *interferir* no estado em superposição sem o medir, o que nos vai permitir retirar uma propriedade global sobre $f(x)$/operador. Mais ainda veremos também como escalar este procedimento para um sistema com mais *qubits*.

4.1.1 Algoritmo Deutsch

A seção anterior, permitiu-nos ganhar intuição sobre o funcionamento do paralelismo quântico. Percebemos para um exemplo conciso, que era possível avaliar *simultaneamente* os dois possíveis resultados de uma determinada função. No entanto, ficamos limitados pela consulta do resultado, uma vez que apenas podemos retirar informação sobre um deles. O algoritmo de *Deutsch* combina a noção de *paralelismo quântico* apresentada anteriormente, com um fenômeno conhecido por *interferência*. O mesmo, vai permitir extrair uma propriedade global sobre a função $f(x)$, anteriormente apresentada. Esta propriedade diz-nos se a função f é *balanceada* isto é, a função devolve um valor para metade do seu *input* e outro para a restante metade, ou então se é *constante*, que como o nome indica devolve o mesmo valor para todo o seu *input*. Vamos começar por apresentar o circuito, analisando da esquerda para a direita e observando as propriedades descritas.

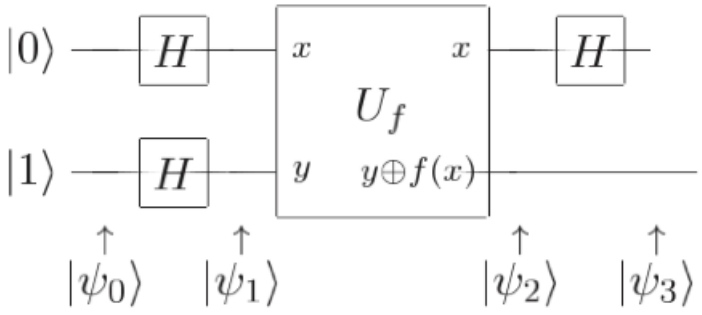


Figura 4.2: Circuito quântico que implementa o algoritmo de *Deutsch's*.

Começando por $|\psi_0\rangle$, facilmente concluímos que,

$$|\psi_0\rangle = |01\rangle. \tag{4.4}$$

Avançando no circuito, chegamos a $|\psi_1\rangle$,

$$|\psi_1\rangle \equiv H^{\otimes 2}|01\rangle = \begin{cases} H|0\rangle \to \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ H|1\rangle \to \frac{|0\rangle-|1\rangle}{\sqrt{2}} \end{cases} \equiv \left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right] \otimes \left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] \equiv |\psi_1\rangle. \tag{4.5}$$

O passo seguinte consiste na aplicação do operador U_f, responsável pela aplicação $|x, y\rangle \to |x, y \oplus f(x)\rangle$. Considere-se a seguinte transformação,

.

$$U_f\left[|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\right] \equiv \frac{1}{\sqrt{2}}|x\rangle[|0\oplus f(x)\rangle - |1\oplus f(x)\rangle] \tag{4.6}$$

$$\equiv \begin{cases} |x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}, sef(x)=0 \\ |x\rangle\frac{(|1\rangle-|0\rangle)}{\sqrt{2}}, sef(x)=1 \end{cases} \tag{4.7}$$

$$\equiv (-1)^{f(x)}|x\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}} \tag{4.8}$$

□

Podemos agora avançar para $|\psi_2\rangle$. Sabemos que $U_f\Big(|x\rangle|y\rangle \to |x\rangle|y\oplus f(x)\rangle\Big)$ e neste momento no nosso circuito temos,

$$U_f\left[\frac{(|0\rangle+|1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\right]. \tag{4.9}$$

Para uma maior legibilidade, vamos optar por realizar o tensor dos dois *qubits*, avançado só depois para a aplicação do operador U_f. Realizando a operação de *produto tensorial*, obtemos,

$$U_f\left(\frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)\right). \tag{4.10}$$

Aplicamos agora o operador U_f, têm-se

$$\frac{1}{2}(|0\rangle|0\oplus f(0)\rangle - |0\rangle|1\oplus f(0)\rangle + |1\rangle|0\oplus f1\rangle - |1\rangle|1\oplus f(1)\rangle) \tag{4.11}$$

$$\frac{1}{2}\left((-1)^{f(0)}|0\rangle\otimes(|0\rangle-|1\rangle) + (-1)^{f(1)}|1\rangle\otimes(|0\rangle-|1\rangle)\right). \tag{4.12}$$

Falta-nos agora analisar as possíveis combinações entre os valores $f(0)$ e $f(1)$, respectivamente, podemos ter,

$$\begin{cases} f(0)=f(1)=0 \\ f(0)=f(1)=1 \\ \\ f(0)=0, f(1)=1 \\ f(0)=1, f(1)=0 \end{cases} \tag{4.13}$$

então fazendo a substituição, vamos obtendo respectivamente,

$$\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{4.14}$$

$$\frac{-(|0\rangle + |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{4.15}$$

$$\frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{4.16}$$

$$\frac{-(|0\rangle - |1\rangle)}{\sqrt{2}} \otimes \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{4.17}$$

ou seja, podemos rescrever o *ouput* como,

$$|\psi_2\rangle \equiv \begin{cases} \pm\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}, \text{ se } f(0) = f(1) \\ \\ \pm\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}, \text{ se } f(0) \neq f(1). \end{cases} \tag{4.18}$$

É nesta fase do circuito, que acontece o fenómeno de *interferência* na *sobreposição*. O seu resultado está codificado no estado $|\psi_3\rangle$. Analisando o circuito verificamos que o operador *Hadamard* é aplicado ao primeiro *qubit* (sobreposição com a avaliação de todos os possíveis valores de $f(x)$). Isto é,

$$\begin{cases} H\left(\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}\right) \equiv |0\rangle \\ \\ H\left(\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}\right) \equiv |1\rangle. \end{cases} \tag{4.19}$$

que pode ser demonstrado por:

$$\begin{aligned} H\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} &\equiv \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle) \equiv \frac{1}{\sqrt{2}}\left(\frac{(|0\rangle + |1\rangle)}{\sqrt{2}} + \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}\right) \\ &\equiv \frac{1}{\sqrt{2}}\frac{2|0\rangle}{\sqrt{2}} \equiv |0\rangle. \end{aligned} \tag{4.20}$$

□

e de forma análoga para a segunda igualdade. Posto isto, podemos definir $|\psi_3\rangle$,

$$|\psi_3\rangle \equiv \begin{cases} \pm|0\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}, \text{ se } f(0) = f(1) \\ \\ \pm|1\rangle\frac{(|0\rangle-|1\rangle)}{\sqrt{2}}, \text{ se } f(0) \neq f(1). \end{cases} \tag{4.21}$$

Se agora, nos apercebermos que $f(0) \oplus f(1)$ é 0 se f(0) = f(1) e 1 caso contrário, então podemos rescrever o estado como,

$$|\psi_{out}\rangle = \pm|f(0) \oplus f(1)\rangle\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right). \tag{4.22}$$

observamos que ao realizarmos uma medição no primeiro *qubit*, obtemos uma propriedade global de f. Isto é se obtivermos 0, **é porque f é constante**, se por outro lado obtivermos 1, **f é balanceada**.

4.1.2 Agortimo Deutsch-Jozsa

Resumidamente, o algoritmo de *Deutsch-Jonza* é a generalização do algoritmo de *Deutsch* para um sistema de *n-qubits*. O algoritmo em si, foi publicado em Deutsch and Jozsa [1992]. A ideia deste passa por se avaliar se uma determinada função f é garantidamente *balanceada* ou *constante*. Pretendemos na mesma, que a avaliação para *n-qubits* da função seja feita simultaneamente. Então, de maneira a comportar os n *qubits*, precisamos de redefinir f como,

$$f(x) : \{0,1\}^n \Longrightarrow \{0,1\}. \tag{4.23}$$

Antes de avançarmos para a avaliação do algoritmo, vamos apresentar uma ideia pitoresca da sua utilização, conhecida por *O problema de Deutsch*. Considere-se o par *Alice* e *Bob*. A *Alice* escolhe um numero x tal que, $2^n - 1 > x > 0$ e envia-o para o *Bob*. Note-se que ela envia n *bits* para *Bob*, os necessário para representar o numero escolhido. A tarefa do *Bob*, é para uma determinada função $f : \{0..2^n-1\} \rightarrow \{0,1\}$, calcular o resultado de $f(x)$ e enviá-lo para a *Alice*, que tem agora de perceber se a função é balanceada ou constante. A pergunta que se coloca é "Qual o custo desta Tarefa?".

Classicamente, a *Alice* apenas envia para *Bob* um x de cada vez (uma mensagem um x). Assim sendo, ela terá que enviar no mínimo $2^n/2 + 1$ perguntas ao *Bob* antes de poder resolver com certeza o problema. Isto deve-se ao facto de ela poder receber $2^n/2$ zeros, antes de finalmente obter um 1, o que lhe diria que a função é balanceada. Convém salientar que o custo da mensagem não faz parte do problema. Apenas é aqui referido para intuitivamente complicar o trabalho de *Bob*. Na realidade este custo pode não existir, sendo por outro lado custoso o calculo de $f(x)$. Por outro lado, recorrendo a computação quântica, a *Alice* e o *Bob* podiam trocar *qubits*. Mais ainda, se o *Bob* realizasse a operação f através do operador unitário U_f, então a Alice conseguiria avaliar o resultado pretendido em apenas uma execução do algoritmo. O mesmo apresenta-se de seguida, recorrendo ao seu circuito quântico Figura 4.3, cuja execução iremos analisar nas próximas linhas. Constatamos que este circuito é muito semelhante ao do algoritmo de *Deutsch*, o que era expectável. Vamos analisá-lo de mesma forma. Começando pelo *input*, $|\psi_0\rangle$, obtemos,

$$|\psi_0\rangle = |0\rangle^{\otimes n}|1\rangle. \tag{4.24}$$

Através do circuito, percebemos que cada um dos n $|0\rangle$, vai ser sujeito ao operador H. No Capítulo sobre os **fundamentos da álgebra linear**, mais concretamente na seção dos postulados sobre sistemas compostos, vimos um resultando teórico que condensava o resultado de aplicarmos o operador H,

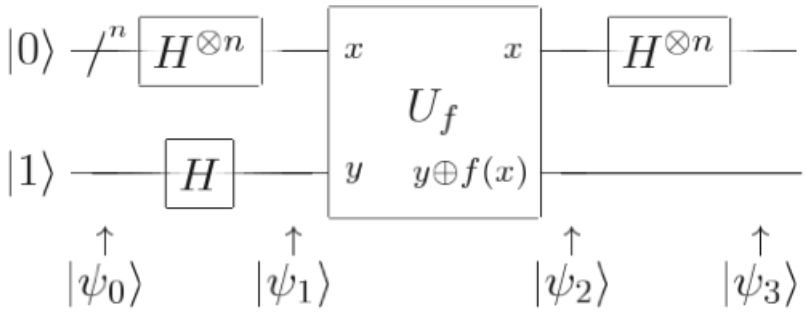


Figura 4.3: Circuito quântico que implementa o algoritmo de *Deutsch-Jozsa*. A notação '/' presente nos cabos/fios quânticos é responsável por representar um conjunto de n *qubits*.

a um sistema de *n-qubits*. Recorrendo por isso a equação sobre o "pós-estado" aí definida, Equação 2.111 podemos representar o estado resultante $|\psi_1\rangle$ como,

$$\sum_{x=0}^{2^n-1} \frac{|x\rangle}{\sqrt{2^n}} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{4.25}$$

Agora o operador U_f, responsável por avaliar a função f, em cada um dos elementos de 0 até $2^n - 1$ vai atuar, deixado o resultado desta avaliação guardado no último registro. Recordando a Equação 4.8, podemos escrever o estado $|\psi_2\rangle$,

$$\sum_{x=0}^{2^n-1} \frac{(-1)^{f(x)}|x\rangle}{\sqrt{2^n}} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{4.26}$$

Neste momento, estamos na "posse" de todos os possíveis $f(x)$ para qualquer $2^n - 1 > x \geq 0$. Esta informação esta toda guardada nos primeiros *n qubits*. Apesar de não lhe conseguirmos aceder diretamente, podemos retirar algum conhecimento geral sobre a mesma, *a dita propriedade global de* $f(x)$. Para o fazemos precisamos de *interferir* na sobreposição. Considerando a simplificação apresentada no *postulado* sobre *sistemas compostos*, nomeadamente a Equação 2.110, que traduz a aplicação do operador *Hadamard* a um sistema de *n qubits*. Obtemos então $|\psi_3\rangle$,

$$|\psi_3\rangle = \sum_k \sum_x \frac{(-1)^{x.k \oplus f(x)}|k\rangle}{2^n} \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right). \tag{4.27}$$

Chegamos a um momento chave do algoritmo. Concentrado-nos no primeiro registo, vamos considerar o caso específico da amplitude do estado $|k\rangle = |0\rangle^{\otimes n}$. Além disso, vamos supor o caso de $f(x)$ ser constante. Neste caso, sabemos que a amplitude do dito estado é dada por, $(-1)^{x.k+f(x)} \equiv \pm 1$. Isto deve-se ao facto de $k = 0$, então $x.k$ também vai ser zero e claro,

$f(x)$ é constante devolvendo 0 ou 1. Ou seja, quando medimos o primeiro registro vamos obter zero. Atendendo também a equação de normalização do estado, que se traduz na soma das probabilidades das amplitudes ser 1, podemos afirmar que quando $f(x)$ é constante, obrigatoriamente obtemos o valor 0 no primeiro registro (após medição).
Se por outro lado f é balanceada, as diferentes contribuições para as amplitudes vão cancelar, deixando uma amplitude de zero. Neste caso uma medição daria um resultado diferente de 0, em pelo menos um *qubit* do primeiro registo. **Resumidamente, se a Alice medir zero em todos os *qubits* do primeiro registo ela sabe que a função é constante, caso contrário a função é balanceada.**

Nas próximas páginas mais algoritmos quânticos serão apresentados. Como teremos oportunidade de observar, todos eles recorrem ao conceito de paralelismo quântico aqui ilustrado, motivo pelo qual conseguem ter um desempenho tão avassalador em comparação com os seus homólogos clássicos conhecidos.

4.2 A Transformada de Fourier

No mundo clássico, a transformada de *Fourier* é normalmente aplicada a problemas de processamento de sinais digitais. Para nós ela é particularmente importante. Esta importância advém da implementação quântica da transformada de *Fourier*, uma vez que ela é o principal ingrediente do algoritmo de *Shor*. Antes de mais é importante salientar que a transformada de *Fourier* quântica não *acelera* a "tarefa" da sua homóloga clássica. Ao invés, ela tira partido da capacidade dos operadores quânticos, realizarem simultaneamente computação em 2^n estados base, usando apenas um sistema de *n-qubits*.

As próximas páginas servirão para enunciar o problema da Transformada de *Fourier* Discreta e a sua mais eficiente implementação, *Fast Fourier Transform*. Em seguida avançaremos para a sua vertente quântica, explorando a mesma com a ajuda de um pequeno exemplo, como temos vindo fazer até agora. Por último, veremos uma aplicação particular da transformada de *fourier* quântica, denominada *estimação de fase*. Resumidamente, ela permite a *aproximação dos valores próprios de um operador unitário em certas condições controladas*. Isto vai permitir resolver alguns problemas intratáveis hoje em dia em computação clássica de forma eficiente.

4.2.1 Transformada de Fourier Discreta

De uma forma simplista, a transformada de *Fourier* não é mais que uma função, que tem como *input* um vector V $\in \mathbb{C}^N$ de tamanho **N** fixo e que devolve um vector Z $\in \mathbb{C}^N$. Ou seja:

$$(v_0, v_1, \ldots, v_{n-1}) \Longrightarrow (z_0, z_1, \ldots, z_{n-1}) \tag{4.28}$$

onde,

$$Z_k = \sum_{j=0}^{N-1} v_j e^{\frac{2\pi i j k}{N}} \tag{4.29}$$

No entanto e por questões relacionadas com a normalização da informação quântica, vamos precisar de uma versão normalizada da transformada de *Fourier* dada por:

$$Z_k = \frac{1}{N} \sum_{j=0}^{N-1} v_j e^{\frac{2\pi i j k}{N}}. \tag{4.30}$$

Este fator de normalização em nada altera as propriedades clássicas da *TFD*, por outro lado, permite-nos implementá-la segundo um algoritmo quântico. Mas antes disso, vamos analisar a ideia básica do algoritmo clássico mais eficiente que se conhece para implementar a **DFT**, chamado **FFT**, *Fast Fourier Transform*:

1. Computa a transformação de um vector de dimensão 2^n.
2. O vector é divido em duas metades iguais, computando recursivamente a **DFT** em cada uma dessas metades.
3. O *Output* resulta da junção das duas metades.
4. O melhor algoritmo clássico tem um peso computacional de $\Theta(N \log N)$. Mas para um vector de dimensão 2^n elementos, traduz-se num custo de cariz exponencial.

4.2.2 Transformada de Fourier Quântica

A *TFQ* é a implementação quântica da transformada de *Fourier* discreta que acabamos de rever. Na realidade ambas representam a mesma transformação, mas no caso da transformada quântica existe uma notação própria. A seguinte apresentação, segue de perto o trabalho de Cheung [2003] assim como a exposição da transformada de *Fourier* presente em Nielsen and Chuang [2000].
Considere-se a base $|0\rangle, \cdots, \left|2^{n-1}\right\rangle$. A TFQ realiza então a seguinte operação em cada elemento da base,

$$TFQ(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{\frac{2\pi i j k}{2^n}} |k\rangle. \tag{4.31}$$

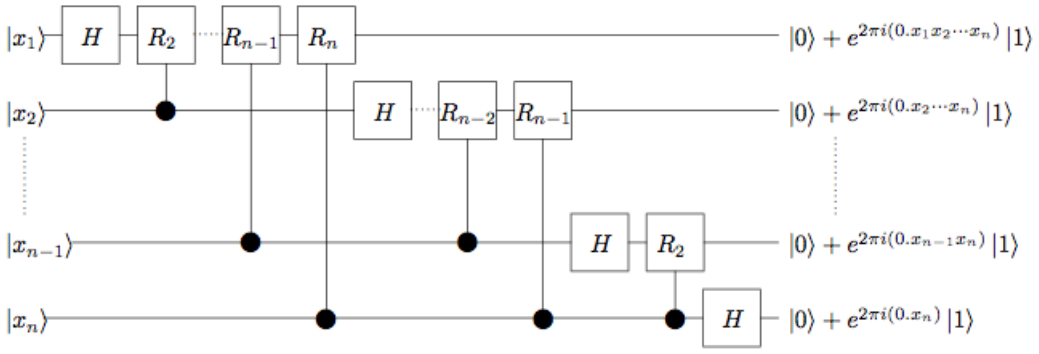


Figura 4.4: Circuito quântico que implementa a transformada de *Fourier* para um sistema de n-qubits

À semelhança da TFD, a transformada de *Fourier* quântica recebe como input um vetor $\mathbb{V} \in \mathbb{C}^n$ isto é,

$$|\psi\rangle = \sum_{j=0}^{2^n-1} v_j|j\rangle, \tag{4.32}$$

onde os elementos v_j correspondem ás amplitudes do vetor/estado de *input*. A transformada de *Fourier* aplicada a esse estado é então:

$$TFQ(|\psi\rangle) = \sum_{j=0}^{2^n-1} TFQ(v_j|j\rangle) = \sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} \frac{v_j e^{\frac{2\pi ijk}{2^n}}}{\sqrt{2^n}}|k\rangle = \sum_{k=0}^{2^n-1} Z_k|k\rangle. \tag{4.33}$$

com Z_k definido como em 4.30. Note-se como a transformada não altera os estados base, mas ao invés modifica as amplitudes associadas a estes. De seguida vamos-nos concentrar na implementação da transformada de *Fourier*.

Considerando a Figura 4.4, é importante salientar três aspectos importantes, para a sua legibilidade. Em primeiro lugar, o estado de *input* $|x\rangle$ é rescrito como o produto do *Tensor* entre os diferentes elementos da base computacional ($|x_1\rangle \otimes |x_2\rangle \ldots \otimes |x_n\rangle$) de *n-qubits*, como é mencionado na Equação 4.32. Além disso, um estado x pode ser representado na sua expansão binária ($x = x_1 2^{n-1} x_2 2^{n-2} \cdots x_n 2^0$), assim como é importante notar a utilização da notação das frações binárias $e^{2\pi i(0.x_l x_{l+1} x_{l+2} \ldots x_{l+m-1})}$ no fim do circuito apresentado, ou seja,

$$0.x_l x_{l+1} x_{l+2} \ldots x_{l+m-1} \equiv \frac{x_l}{2} + \frac{x_{l+1}}{4} + \frac{x_{l+2}}{8} + \frac{x_{l_{+m-1}}}{2^m} \tag{4.34}$$

Em segundo lugar, as operações R_m traduzem-se numa rotação de fase controlada, mais concretamente é adicionado uma fase relativa à componente $|1\rangle$

do estado isto é,

$$R_m \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^m}} \end{bmatrix}. \tag{4.35}$$

Por último, para obtermos verdadeiramente a TFQ pelo dito circuito é necessário inverter a ordem final dos *qubit*[1], obtendo-se o seguinte *output*:

$$\frac{(|0\rangle + e^{2\pi i(0.x_n)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(0.x_{n-1}x_n)}|1\rangle) \otimes \ldots \otimes (|0\rangle + e^{2\pi i(0.x_1x_2\ldots x_n)}|1\rangle)}{\sqrt{2^n}} \tag{4.36}$$

Por forma a corroborar que este resultado corresponde a Equação 4.31, vamos transcrever da obra de Cheung [2003][Página 22-23] a dita prova:

. Tendo em atenção que $e^{2\pi i} = 1$

$$e^{2\pi i(0.x_1x_2\ldots x_n)} \equiv e^{2\pi i(x_1x_2\ldots x_n)} \equiv e^{2\pi i(2^{n-1}x/2^n)} \tag{4.37}$$

isto permite-nos rescrever o estado como,

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i(2^{n-1}x/2^n)}|1\rangle) \otimes \ldots \otimes (|0\rangle + e^{2\pi i(2^1x/2^n)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^0x/2^n)}|1\rangle) \tag{4.38}$$

se considerarmos agora que o nosso estado base é dado por *n-qubit* $|k\rangle = |k_1k_2\ldots k_n\rangle$, podemos mais uma vez redefinir o output do circuito como,

$$\frac{1}{\sqrt{2^n}}(e^{2\pi ik_1(2^{n-1}x/2^n)}|k_1\rangle) \otimes \ldots \otimes (e^{2\pi ik_{n-1}(2^1x/2^n)}|k_{n-1}\rangle) \otimes (e^{2\pi ik_n(2^0x/2^n)}|k_n\rangle) \tag{4.39}$$

usando agora o facto de $e^{2\pi i(0)} = 1$ e expandindo a operação do produto tensorial obtemos:

$$\begin{aligned} &\frac{1}{\sqrt{2^n}}\left(e^{2\pi ik_1(2^{n-1}x/2^n)}\right)\ldots\left(e^{2\pi ik_{n-1}(2^1x/2^n)}\right)\left(e^{2\pi ik_n(2^0x/2^n)}\right)|k_n\rangle \\ &\quad\equiv\frac{1}{\sqrt{2^n}}e^{2\pi i(2^{n-1}k_1+\ldots+2^1k_{n-1}+2^0k_n)(x/2^n)}|k\rangle \\ &\quad\equiv\frac{1}{\sqrt{2^n}}e^{2\pi ikx/2^n}|k\rangle. \end{aligned} \tag{4.40}$$

agora só falta somar todas as 2^n componentes, isto é,

$$\frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{2\pi ikx/2^n}|k\rangle = QFT(|x\rangle).^2 \tag{4.41}$$

□

[1] este processo não está representado na figura

[2] QFT - do inglês *quantum Fourier transform.*

Em seguida e para melhor sedimentarmos o assunto, vamos apresentar um exemplo conciso. Vamos então calcular a transformada de *Fourier* num sistema de *2-qubits* e apresentaremos o seu circuito, fazendo uma analise do mesmo. Uma vez que possuímos *2-qubits* então,

$$N = 2^2 \equiv 4. \tag{4.42}$$

Pela equação 4.31, podemos escrever,

$$|j\rangle \longrightarrow \frac{1}{2}(e^{2\pi i\times 0}|0\rangle + e^{\frac{2\pi i\times 1\times j}{4}}|1\rangle + e^{\frac{2\pi i\times 2\times j}{4}} + |2\rangle + e^{\frac{2\pi i\times 3\times j}{4}}|3\rangle) \tag{4.43}$$

onde os estados são representados por:

$$|0\rangle = |00\rangle \tag{4.44}$$
$$|1\rangle = |01\rangle \tag{4.45}$$
$$|2\rangle = |10\rangle \tag{4.46}$$
$$|3\rangle = |11\rangle \tag{4.47}$$

então, pela equação 4.33, temos agora que escrever explicitamente a transformada para cada *qubit* $|j\rangle$(elemento da base),

$$|0\rangle \Longrightarrow \frac{1}{2}(e^{\frac{2\pi i\times 0}{4}}|0\rangle + e^{\frac{2\pi i\times 0}{4}}|1\rangle + e^{\frac{2\pi i\times 0}{4}}|2\rangle + e^{\frac{2\pi i\times 0}{4}}|3\rangle) \tag{4.48}$$
$$|1\rangle \Longrightarrow \frac{1}{2}(|0\rangle + e^{\frac{2\pi i\times 1\times 1}{4}}|1\rangle + e^{\frac{2\pi i\times 2\times 1}{4}}|2\rangle + e^{\frac{2\pi i\times 3\times 1}{4}}|3\rangle \tag{4.49}$$
$$|2\rangle \Longrightarrow \frac{1}{2}(|0\rangle + e^{\frac{2\pi i\times 1\times 2}{4}}|1\rangle + e^{\frac{2\pi i\times 2\times 2}{4}}|2\rangle + e^{\frac{2\pi i\times 3\times 2}{4}}|3\rangle \tag{4.50}$$
$$|3\rangle \Longrightarrow \frac{1}{2}(|0\rangle + e^{\frac{2\pi i\times 1\times 3}{4}}|1\rangle + e^{\frac{2\pi i\times 2\times 3}{4}}|2\rangle + e^{\frac{2\pi i\times 3\times 3}{4}}|3\rangle \tag{4.51}$$

Então podemos construir o operador que implementa a **TFQ** para este sistema de *2-qubits*. Ou seja a transformação:

$$|j\rangle = F|k\rangle \tag{4.52}$$

onde,

$$F \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & . & . & . \\ 1 & . & . & . \\ 1 & . & . & . \end{bmatrix} \tag{4.53}$$

para preenchermos o resto do operador basta-nos olhar para a transformada, considerando cada elemento da base $(|00\rangle, |01\rangle, |10\rangle, |11\rangle)$ como uma coluna, como temos vindo a fazer, copiando em seguida os valores das amplitudes/-coeficientes associadas para cada linha. Como exemplo temos o elemento $|0\rangle$. No desdobramento da transformada a base $|0\rangle$ tem sempre coeficiente **1**,

motivo pelo qual na primeira coluna de **F** ($|00\rangle$) todos os elementos tem esse valor. Procedendo da mesma forma com os restantes elementos obtemos:

$$F \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \tag{4.54}$$

No capítulo anterior, já tínhamos visto este operador, utilizando por nós para ilustrar o algoritmo de **unitários de nível 2**. De seguida apresentaremos o circuito que implementa esta transformada. Mas antes, e devido ao último *passo* do mesmo, a inversão dos *qubits*, aproveitamos para apresentar um operador muito simples, $SWAP$ para esse efeito. Em termos lógicos queremos inverter $|a, b\rangle$ para $|b, a\rangle$, então:

$$|a, b\rangle \Longrightarrow |a, a \oplus b\rangle \tag{4.55}$$
$$\longrightarrow |a \oplus (a \oplus b), a \oplus b\rangle \equiv |b, a \oplus b\rangle \tag{4.56}$$
$$\longrightarrow |b, (a \oplus b) \oplus b\rangle \equiv |b, a\rangle. \tag{4.57}$$

Isto é conseguido pelo seguinte circuito:

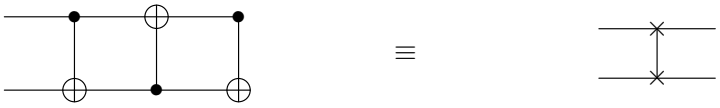


Figura 4.5: Circuito do operador $SWAP$ - que troca dois *qubits*.

O operador $SWAP$, tem então a seguinte representação matricial:

$$SWAP \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.58}$$

Posto isto, apresentamos agora o circuito que implementa a transformada de *Fourier* para um sistema de *2-qubits*

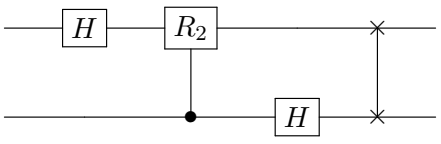


Figura 4.6: Transformada de *Fourier* quântica, para um sistema de *2-qubits*

É fácil de analisar a relação entre os qubit de *input* e os qubit *output*. A título de exemplo, consideremos o estado $|00\rangle$. Como sabemos ele pode ser

representado pelo vetor tal que,

$$|00\rangle \equiv \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.59}$$

Aplicando o operador **F**, obtemos, como seria de esperar, a transformada de *Fourier* para o estado $|0\rangle$.

$$F|00\rangle \equiv \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \bigotimes \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \equiv \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \equiv \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle). \tag{4.60}$$

A ação nos restantes elementos da *base* é similar, motivo pelo qual não a apresentamos aqui. Vamos agora analisar a complexidade desta operação. Como foi dito inicialmente, o melhor algoritmo clássico, **Fast Fourier Transform** consegue computar a transformada em tempo exponencial. Analisando a implementação da **TFQ** verificamos que começamos por realizar a operação **H**, seguida de *n-1* rotações, isto perfaz *n-operações* no primeiro *qubit*. Avançando para o segundo *qubit*, temos **H** seguido de *n-2* rotações. Neste momento já temos um total de *n+(n-1)* operadores. Continuando esta linha de pensamento, verificamos que precisamos de $n+(n-1)+\ldots+1 = n(n+1)/2$ operações. A isto ainda temos de juntar *n/2 SWAPS*. Em suma este circuito permite uma computação da transformada de *Fourier* em $\Theta(n^2)$, em contraste com a sua homóloga clássica, $\Theta(n2^n)$. Motivo mais que suficiente para justificar o *frenesim* em volta desta abordagem.

4.2.3 Estimação de Fase Quântica

Vamos agora analisar uma aplicação que faz uso da transformada de *Fourier* como uma sub-rotina, conhecida por *estimação de fase*. A seguinte exposição segue de perto os trabalhos de Nielsen and Chuang [2000] e Cheung [2003]. O problema é-nos apresentado nos seguintes moldes. Seja **U** um operador unitário nosso conhecido, assim como um seu vector próprio $|v\rangle$. Relembrando o Capítulo 2, foi dito que era possível *decompor* a ação de um operador num estado como,

$$U|v\rangle = \lambda|v\rangle, \text{ onde } \lambda \text{ é o valor próprio respetivo .} \tag{4.61}$$

este escalar λ, iria alterar a *magnitude* do vector, mantendo a direção *paralela* em relação ao original. Voltando novamente ao problema em mãos, este algoritmo procura então estimar λ, sabendo que este é da forma:

$$\lambda = e^{2\pi i\theta}. \tag{4.62}$$

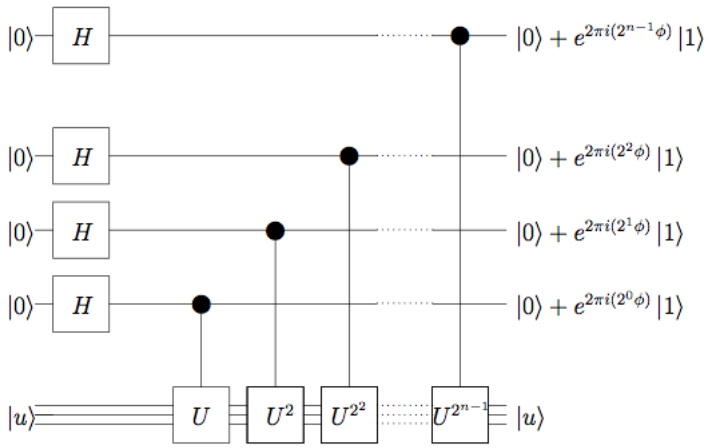


Figura 4.7: Primeira parte do algoritmo de estimação de fase.

Vamos considerar que θ é um múltiplo de $\frac{1}{2^n}$(n corresponde ao número de *qubits* auxiliares, como se apresenta mais abaixo. Este será codificado com a sintaxe apresentada na seção da transformada de *Fourier*, nomeadamente

$$\theta = (0.x_1x_2\ldots x_n). \tag{4.63}$$

Antes de avançarmos mais, convém salientar que este algoritmo pressupõe a utilização de dois registos isto é, dois *inputs*. O primeiro, *auxiliar*, é usado para estabelecer com a precisão que queremos obter na estimação da fase, daí a palavra *estimação*. O segundo, *carrega* o estado propriamente dito. Vamos agora apresentar o circuito quântico deste algoritmo. Por motivos de legibilidade, vamos dividi-lo duas partes, respectivamente Fig. 4.7 e Fig. 4.8, fazendo uma análise do seu funcionamento. A aplicação concreta deste algoritmo com um problema prático, será observada no próximo capitulo. O circuito presente na figura 4.7, implementa a primeira parte deste procedimento. Analisando-o da esquerda para a direita verificamos:

1. O primeiro *registo*, contém **n** *qubits* inicialmente do estado $|0\rangle$. Estes são os *qubits* auxiliares. Deles dependem duas coisas, o número de dígitos usados (precisão), e qual a probabilidade de sucesso que queremos que o algoritmo tenha.

2. O segundo *registro*, começa no estado $|u\rangle$ e contém os *qubits* necessários para representar o estado $|u\rangle$.

3. O circuito começa por aplicar ao primeiro *registo*, o operador **H**, deixando cada *qubit* no estado $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Em seguida a operação, U^{2^j} controlada é aplicada ao vetor próprio-$|u\rangle$. Ou seja, se o *input* for $|0\rangle|u\rangle$

o operador não tem efeito e o *output* é $|0\rangle|u\rangle$. Caso contrário, isto é $|1\rangle|u\rangle$, a operação devolve:

$$\begin{aligned} c - U^{2^j}(|1\rangle|u\rangle) &\equiv |1\rangle(U^{2^j}|u\rangle) \\ &\equiv |1\rangle(e^{2\pi i(2^j\theta)}|u\rangle) \equiv e^{2\pi i(2^j\theta)}|1\rangle|u\rangle \\ &\equiv (e^{2\pi i(2^j\theta)}|1\rangle)|u\rangle. \end{aligned} \tag{4.64}$$

4. Posto isto, é facil de perceber, que dado o *input* das operações *Controlled-*U^{2^j} ser $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. As mesmas vão devolver, $\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(2^j\theta)}|1\rangle)$, o que perfaz exactamente o que obtemos no final da primeira parte do circuito.

Ao analisarmos o *output* do circuito constatamos que o estado final é dado por:

$$\big|\psi'\big\rangle = \frac{(|0\rangle + e^{2\pi i(2^{n-1}\theta)}|1\rangle) \otimes (|0\rangle + e^{2\pi i(2^1\theta)}|1\rangle) \otimes \ldots \otimes (|0\rangle + e^{2\pi i(2^0\theta)}|1\rangle)}{\sqrt{2^n}} \tag{4.65}$$

Relembrando o estado final da transformada de *Fourier*, Equação 4.36, assim como, que a fase θ é um múltiplo exacto de $1/2^n$, codificado como a *fração binaria*, $\theta = (0.x_1x_2\ldots x_n)$. Verificamos que ambos os *outputs* são iguais. Ou seja, *intuitivamente*, se aplicarmos a *inversa* da **TQF**, ou seja, TQF^{-1}, o *output* será $|x_1x_2\ldots x_n\rangle$, que não é mais do que os *bits* individuais da *representação binaria* de θ.
A segunda parte do algoritmo de estimação de fase traduz-se portanto na aplicação da inversa da transformada de *Fourier*. Como?. Precisamos de mais um conceito de *álgebra linear*. Da *álgebra linear*, sabemos que a inversa do produto de operadores unitários, consiste, no produto das inversas desses operadores por ordem inversa, isto é:

$$(U_1U_2\ldots U_n)^{-1} \equiv U_n^{-1}\ldots U_2^{-1}U_1^{-1}. \tag{4.66}$$

Posto isto, podemos construir trivialmente a inversa do circuito da transformada de *Fourier*, que é aplicada aos primeiros *n qubits – primeiro registo*(Figura 4.7). Sabemos que o estado de *input* deste circuito é dado pela Equação 4.65. Podemos agora rescrever o mesmo, definindo os n estados base por, $|0\rangle, |1\rangle, |2\rangle, \ldots, |2^n - 1\rangle$ e expandindo o produto tensorial, ou seja,

$$\big|\psi'\big\rangle \equiv \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i(k\theta)}|k\rangle. \tag{4.67}$$

Se agora aplicarmos a inversa de TQF, responsável por transformar cada estado base $|k\rangle$ em,

$$TFQ^{-1}(|k\rangle) = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi ijk/2^n}|j\rangle \tag{4.68}$$

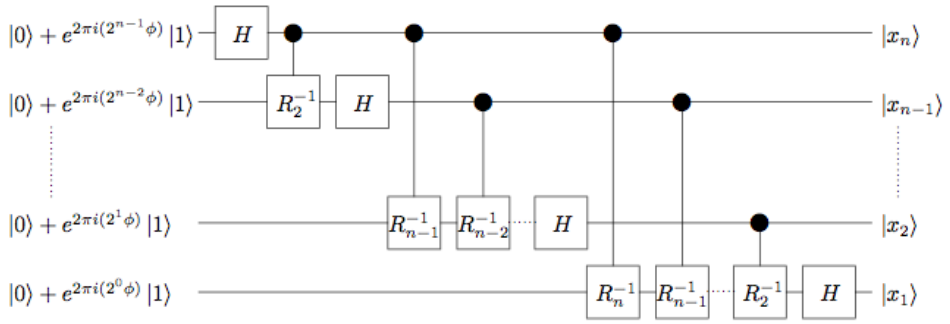


Figura 4.8: Circuito da inversa da transformada de *Fourier* TFQ^{-1}, aplicada aos primeiros *n-qubits*

ao nosso estado $|\psi'\rangle$, obtemos

$$TFQ(|\psi'\rangle) = \frac{1}{2^n}\sum_{j=0}^{2^n-1}\sum_{k=0}^{2^n-1} e^{-2\pi ijk/2^n} e^{2\pi i(k\theta)}|j\rangle. \tag{4.69}$$

Deste resultado podemos inferir que a amplitude *particular* de cada estado base $|j\rangle$ é dada por,

$$\alpha_j = \frac{1}{2^n}\sum_{k=0}^{2^n-1} e^{-2\pi ijk/2^n} e^{2\pi i(k\theta)} = \frac{1}{2^n}\sum_{k=0}^{2^n-1} e^{2\pi i(\theta - j/2^n)k}, \tag{4.70}$$

que é uma série geométrica. Assim sendo, dado que $\theta = j/2^n$, que não é mais do que um múltiplo de $1/2^n$, sabemos que

$$a_j = \frac{1}{2^n}\sum_{k=0}^{2^n-1} 1 = 1. \tag{4.71}$$

Este resultado, vai-nos permitir usar o procedimento de estimação de fase para descobrir a ordem de uma função periódica, passo fundamental no algoritmo de fatorização, que será alvo de estudo no próximo capitulo. Mais ainda, pela análise da série geométrica, podemos afirmar que mesmo no caso de θ não ser um múltiplo exato de $j/2^n$ é possível obter uma boa aproximação a este valor. A prova deste resultando encontra-se em, Nielsen and Chuang [2000]-pag.224].

Graças a ela, podemos definir o número de *qubits auxiliares* no primeiro registo, necessários a estimação de θ com n *bits* de precisão. Mais ainda, afirma-se que a probabilidade de sucesso é de $1-\epsilon$. O número de *qubits auxiliares* é então dado por,

$$t = n + [\log(2 + \frac{1}{2\epsilon})]. \tag{4.72}$$

De uma forma geral, podemos representar esta rotina num único circuito, apresentado na Figura 4.9.

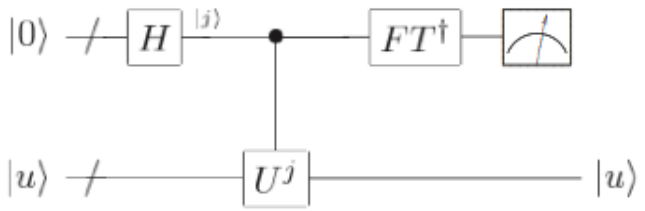


Figura 4.9: Visão geral da rotina de estimação da fase. O simbolo '/' omite o numero necessario de *qwires* para representar os *qubits*, tanto ao nível do primeiro registro como do segundo. $|u\rangle$ é um *vetor próprio* de U com *valor próprio* $e^{2\pi i\varphi}$

4.3 Sumário

Neste capítulo tivemos o primeiro contato com algoritmos quânticos. A escolha da amostra apresentada, incidiu sobre os algoritmos cuja "performance" se sabe ser mais elevada, face aos seus *homólogos clássicos*. Resumidamente, podemos afirmar que os algoritmos quânticos baseados nas propriedades da transformada de *Fourier*, apresentam normalmente reduções de tempos de computação de ordem exponencial, face aos melhores algoritmos clássicos conhecidos. Esta capacidade, provém do conceito de *paralelismo quântico*, abordado também neste capítulo.
No próximo capítulo, apresentaremos provavelmente o algoritmo quântico mais conhecido - *Algoritmo de Shor*. Para tal vamos recorrer a um exemplo - a fatorização do número 15.

Capítulo 5

O Algoritmo de Shor

Quantum mechanics: Real Black Magic Calculus.

Albert Einstein

Em 1997, **Peter Shor** apresentou um artigo Shor [1997] onde figuravam dois algoritmos quânticos. Esses dois algoritmos implementavam respectivamente, a *factorização de números inteiros* e o *Logaritmo Discreto*. Este capítulo tem como mote o estudo do algoritmo de fatorização proposto por *Peter Shor*. Este esforço da nossa parte é motivado pelo impacto deste algoritmo nos esquemas *criptograficos* em uso hoje em dia. Atualmente, um dos sistemas criptográficos mais usados no mundo inteiro é o *RSA*. O cerne deste, prende-se com o custo, nos computadores atuais, de fatorizar números muito grandes. Concretamente, o melhor algoritmo que se conhece para o efeito, tem um custo de cariz *exponencial*, dado por, $\Theta(\exp((\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}))$. O algoritmo de *Shor* por outro lado, consegue resolver este problema, em tempo *polinomial*, $\Theta(\log N)^2(\log\log N)(\log\log\log N))$. [1]
Se conseguíssemos construir um computador quântico capaz de correr este algoritmo, seríamos capazes de aceder rápida e eficazmente a um conjunto muito grande de informação protegida.

Posto isto, vamos começar por fazer uma revisão sobre algumas propriedades da *Teoria dos Números*, que são relevantes para o algoritmo de fatorização. De seguida, veremos a estratégia que o algoritmo proposto por *Peter Shor* seguiu, que consiste em reduzir o problema da fatorização, no problema quântico de *encontrar a ordem* de um elemento, que recorre ao algoritmo de estimação de fase, analisado no capítulo anterior. Finalmente apresentaremos um exemplo concreto, *a factorização do número 15*, para consolidar todos estes conceitos.

[1] Valores retirados do artigo citado.

5.1 Conceitos de Teoria dos Números

Nesta seção apresentaremos os fundamentos necessários à compreensão do algoritmo de fatorização proposto por Peter Shor. Começaremos por rever alguns conceitos sobre teoria dos números, acompanhando-os sempre que possível de pequenos exemplos. Mais ainda, apresentaremos o problema clássico de fatorização como caso prático, salientando as dificuldades clássicas do mesmo. Para aumentar a legibilidade da seção, optou-se por deslocar para a seção dos anexos A, a prova de alguns dos conceitos aqui referidos. A seguinte discussão baseia-se em Ekert and Jozsa [1996] e Ouellette [2002].

Grosso modo, a ideia geral do algoritmo de *Shor* é a seguinte. Dado um N que pretendemos fatorizar e um valor y escolhido de forma aleatória, mas que seja co-primo de N isto é, gcd(y,N) = 1(Anexo A.1), pretende-se descobrir a ordem r da seguinte função,

$$F_N(a) = y^a \mod N. \tag{5.1}$$

este é o ponto crítico do algoritmo clássico isto é, o motivo pelo qual para valores de N muito grandes o mesmo se torna intratável. Veremos nas próximas páginas o porquê da ligação entre o problema da ordem e da fatorização.

Teorema 3. Suponhamos que N é um número composto e x é uma solução não trivial da equação,

$$x^2 = 1(\mod N), \tag{5.2}$$

no domínio $1 \leq x \leq N$. Então pelo menos um dos valores $gcd(x-1, N)$ ou $gcd(x+1, N)$ é um fator não trivial de N.

Demonstração. Vamos começar por rescrever $x^2 \equiv 1 \mod N$ como $x^2 - 1 \equiv 0 \mod N$. Sabemos que $x^2 - 1 = (x+1)(x-1)$, o que implica que N divide $(x+1)(x-1)$. Para isto ser verdade, então N deve ter fatores comuns com $(x+1)$ ou $(x-1)$. Pelo enunciado do teorema também excluímos x de ter uma solução trivial, logo

$$1 < x < N - 1. \tag{5.3}$$

o que nos leva a concluir que,

$$x - 1 < x + 1 < N, \tag{5.4}$$

ou seja, excluímos N de ser o fator comum. Nesse caso o mesmo só pode advir do cálculo de $gcd(x+1, N)$ ou $gcd(x-1, N)$. □

Vamos considerar um exemplo. Considere-se o número composto 341 $(341 = 11 \times 33)$. Como solução não trivial da Equação (5.2), temos,

$$x \equiv \pm 32 \mod 341. \tag{5.5}$$

aplicando agora o algoritmo de *Euclids* à solução x proposta, facilmente obtemos os fatores de N,

$$gcd(31, 341) = 31 \tag{5.6}$$
$$gcd(33, 341) = 11. \tag{5.7}$$

O passo crítico do algoritmo é então descobrir uma solução não trivial x. Como? Se dado um N, escolhermos um y aleatório tal que $y < N$ e se y e N forem co-primos, então definimos r como a ordem de y modulo N(Anexo A.2). Isto é precisamente o período de $F_N(a)$ (equação 5.1). Assim,

$$y^r \equiv 1 \mod N. \tag{5.8}$$

Se r for par, então aplicamos,

$$x = y^{r/2}, \tag{5.9}$$

obtendo $x^2 \equiv 1 \mod N$, o que faz de x um candidato a uma solução não trivial da equação 5.2. Este raciocíno produz a ligação entre a periodicidade da função $F_N(a)$ e o calculo de um fator não trivial de N. No entanto este procedimento pode falhar se o valor y escolhido tiver uma ordem r ímpar ou então se tivermos o "azar" de $y^{r/2}$ ser uma solução trivial. O próximo teorema exprime a probabilidade destes acontecimentos:

Teorema 4. *Seja N um número impar com a seguinte fatorização em primos,*

$$N = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}. \tag{5.10}$$

Supondo que y é escolhido aleatoriamente, tal que gcd(y,N) = 1. Sendo r a ordem de y mod N. Então,

$$Prob(r \text{ é par e } y^{r/2} \neq \pm 1 \mod N) \geq 1 - \frac{1}{2^{k-1}}. \tag{5.11}$$

Demonstração. Nielsen and Chuang [2000] [pag-634]. □

Como exemplo, vamos fatorizar o número 15 usando os conceitos aqui apresentados.

1. Primeiro escolhemos um $y < N$, tal que, $gcd(y, N) = 1$, isto é, $y \in \{2, 4, 7, 8, 11, 13, 14\}$.

2. Vamos escolher o 11. Agora computámos a ordem de 11 mod 15. Ou seja valores, $11^a \mod 15$, para $a = \{1, 2, 3, \dots\}$ até que seja obtido. Isto vai produzir o respectivo resultado $11, 1, 11, 1, 11, \dots$, dando $r = 2$.

3. A seguir computámos $x = y^{r/2}$, o que devolve x = 11. O próximo passo é então calcular o maior fator comum, $gcd(11 \pm 1, N)$. Ou seja, $gcd(10, 15) = 5$ e $gcd(12, 15) = 3$. Estes correspondem à fatorização do número 15.

O procedimento demonstrado convergia para a solução correta com todos os valores de $y \in \{2, 4, 7, 8, 11, 13, 14\}$, menos com o valor 14. Se a escolha aleatória do y caísse no elemento 14, obteríamos $r = 2, y^{r/2} \equiv -1 \mod 15$, o que era uma solução trivial. O Algoritmo de *Shor's* combina então a noção de paralelismo quântico, juntamente com um algoritmo quântico eficiente, capaz de encontrar a ordem de uma função $F_n(a)$. Concretamente isto é e conseguido afinando o algoritmo de estimação de fase quântica, como veremos na próxima seção.

5.2 O problema da Ordem

Como foi dito na seção anterior, o problema de encontrar a ordem recai no problema de estimação de fase já estudado. Para o mesmo precisamos de satisfazer 2 requisitos. Em primeiro lugar, temos que ter um método capaz de implementar uma operação controlada U do tipo U^{2^j}, para um qualquer inteiro j, de maneira a estas operações implementarem o algoritmo de encontrar a ordem. O segundo requisito é sermos capazes de preparar um vetor próprio $|u_s\rangle$, com um valor próprio não trivial, ou pelo menos a *superposição* desses estados. Começando pelo primeiro requisito, ele é satisfeito recorrendo ao método da *exponenciação modular*. A transformação que o mesmo descreve encontra-se de seguida,

$$|z\rangle|y\rangle \Longrightarrow |z\rangle U^{z_t 2^{t-1}} \dots U^{z_1 2^0}|y\rangle \tag{5.12}$$

$$\equiv |z\rangle \left| x^{z_t^{2^{t-1}}} \times \dots \times x^{z_1 2^0} y(\mod N) \right\rangle \tag{5.13}$$

$$\equiv |z\rangle|x^z y(\mod N)\rangle. \tag{5.14}$$

Resumidamente, esta traduz a propriedade de a sequência de operações controladas U^{2^j} usadas no algoritmo de estimação de fase ser equivalente a multiplicar o segundo registo pela exponenciação modular $x^z(\mod N)$, onde z é o conteúdo do primeiro registo.
O segundo requisito é mais intrincado. Preparar $|u_s\rangle$ implica conhecermos r, o que não é possível, uma vez que,

$$|u_s\rangle \equiv \frac{1}{r} \sum_{k=0}^{r-1} e^{\frac{-2\pi i s k}{r}} \left| x^k \mod N \right\rangle, \tag{5.15}$$

para qualquer $s \in [0 \dots r-1]$, corresponde aos vetores próprios de U. No entanto podemos recorrer ao facto de,

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle.^{2} \tag{5.16}$$

Vamos então definir a transformação unitária U_f, responsável por executar a operação de exponenciação modular,

$$U_f|x\rangle|1\rangle = |x\rangle|f(x)\rangle, \tag{5.17}$$

onde,

$$f(x) = y^x \mod N, \tag{5.18}$$

em que y é escolhido aleatoriamente de entre os coprimo de N ou seja, $gcd(y, N) = 1$ e possui uma representação de *L-bits*. Começando o algoritmo de estimação de fase, precisamos de inicializar os dois registos $|0\rangle^{\otimes t}$ e $|1\rangle^{\otimes L}$,

$$|\psi_0\rangle = |0\rangle^{\otimes t}|1\rangle^{\otimes L}. \tag{5.19}$$

O primeiro passo do circuito é aplicar ao primeiro registo o operador $H^{\otimes t}$, obtendo-se

$$H^{\otimes t}|\psi_0\rangle = \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|1\rangle^{\otimes L} \equiv |\psi_1\rangle. \tag{5.20}$$

Este é o momento onde aplicamos o operador U_f ao estado $|\psi_1\rangle$,

$$U_f|\psi_1\rangle = \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|f(x)\rangle \tag{5.21}$$

$$\equiv \frac{1}{\sqrt{2^t}} \sum_{x=0}^{2^t-1} |x\rangle|y^x \mod N\rangle. \tag{5.22}$$

O próximo passo é realizar uma medição na base computacional no segundo registo para determinar os valores dos *bits*. Vamos supor que obtemos o valor z onde $z = y^l \mod N$ para o menor l possível. Então se r é a ordem de y mod N, $y^l \equiv y^{jr+l} \mod N$ para todos os j. Vamos analisar este passo de uma forma mais intuitiva. Já percebemos que o primeiro registo contém a superposição de todos os estados possíveis, enquanto que o segundo contém o cálculo do período da função f para cada um dos valores em superposição. Ou seja, **a medição vai selecionar todos os x's do primeiro registo que contém o mesmo período obtido na medição**. Respectivamente, $x = l, l+r, l+2\times r, \dots, l+Kr$, onde K é o maior inteiro menor que $(2^t - l)/r$.

[2] É possível consultar esta demonstração no anexo A.4

Note-se que $l \leq r \leq N$ e como tal $K \approx 2^t/r$. Voltando à formulação e incorporando estas ideias, definimos o estado pós-medição como,

$$|\psi_3\rangle = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |l + kr\rangle |z\rangle. \tag{5.23}$$

Ficamos então com uma sobreposição particular de estados base, onde cada estado deste conjunto foi selecionado com período r. Queremos agora extrair dessa informação, o r. Nesta fase do procedimento podemos alienar o segundo registo, considerando apenas o primeiro. Para extrair a informação sobre o período r codificada, vamos precisar de aplicar uma TFD_{2^t} a cada um dos estados "selecionados". Isto é, para cada estado,

$$|kr + l\rangle \rightarrow QFT_{2^t} \equiv \frac{1}{\sqrt{2^t}} \sum_{u=0}^{2^t-1} e^{\frac{2\pi i u(kr+l)}{2^t}} |u\rangle. \tag{5.24}$$

Aplicando isto a cada um dos estados definidos em $|\psi_3\rangle$ obtemos,

$$QFT_{2^t}|\psi_3\rangle \equiv \sum_{u=0}^{2^t-1} \left[\frac{1}{\sqrt{2^t K}} \sum_{k=0}^{K-1} e^{\frac{2\pi i(kr+l)u}{2^t}} \right] |u\rangle. \tag{5.25}$$

$$\equiv \sum_{u=0}^{2^t-1} \left[\frac{1}{\sqrt{2^t K}} \left(\sum_{k=0}^{K-1} e^{\frac{2\pi i k r u}{2^t}} \right) e^{\frac{2\pi i l u}{2^t}} \right] |u\rangle. \tag{5.26}$$

Considerando este estado,

$$\sum_{u=0}^{2^t-1} \left[\frac{1}{\sqrt{2^t K}} \sum_{k=0}^{K-1} e^{\frac{2\pi i k r u}{2^t}} \right] e^{\frac{2\pi i l u}{2^t}} |u\rangle, \tag{5.27}$$

e pelo trabalho de Ekert and Jozsa [1996], afirmamos que o termo entre parênteses retos é zero[3], caso u não seja múltiplo de $2^t/r$. Por outro lado o outro termo iguala $2^t/r$. Ou seja, a transformada de *Fourier* de um estado com período r, é um estado com período $2^t/r$. Rescrevendo, u como $j2^t/r$, obtemos,

$$F_{2^t}|\psi_3\rangle \equiv \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{\frac{2\pi i j l}{r}} \left| j\frac{2^t}{r} \right\rangle. \tag{5.28}$$

Ou seja, uma medição no estado u, vai proporcionar um múltiplo $j2^t/r$ com $j = 0, \ldots, r-1$, de forma provável. Isto é, a dita medição retorna um u, tal que,

$$\frac{u}{2^t} = \frac{\lambda}{r}, \tag{5.29}$$

onde u e 2^t são conhecidos. Então usando o algoritmo das *frações continuas*(Anexo A.3) podemos determinar o período r.

[3]pela sua decomposição binaria - $i \Rightarrow \sum_{l=0}^{L-1} 2^l i_l$

Apesar de termos concentrado os nossos esforços em perceber como estimamos r, quando u é um múltiplo de $2^t/r$, esta não é a única situação onde o algoritmo consegue estimar r com uma boa probabilidade. Segundo o trabalho de *Peter Shor* em Shor [1997], a única restrição que o mesmo têm na estimação de r é que u respeite,

$$-\frac{r}{2} \leq ru \mod 2^t \leq \frac{r}{2}, \tag{5.30}$$

o que é equivalente para um determinado d a,

$$\|ru - d2^t\| \leq \frac{r}{2}. \tag{5.31}$$

Se agora re-arranjarmos a expressão obtemos,

$$\left\| \frac{u}{2^t} - \frac{d}{r} \right\| \leq \frac{1}{2 \times 2^t}. \tag{5.32}$$

onde mais um vez estamos na posse de u e 2^t, permitindo-nos o algoritmo das frações continuas extrair r. Desde que o período seja par, sabemos que pelo menos um dos $gcd(y^{r/2} \pm 1, N)$ é um fator não trivial de N. Caso r seja impar, o algoritmo falha e temos que tentar outra vez com um diferente y.

5.3 Shor - Factorização do 15

Nesta seção vamos como o título indica aplicar o algoritmo de fatorização de Shor. Procuramos abordar todos os passos do algoritmo bem como, apresentar na Figura 5.1 o respectivo circuito quântico (genérico).
Vamos começar por definir N = 15. Pelo algoritmo clássico, sabemos que temos de escolher um y tal que, $y \leq N$ e $gcd(y, N) = 1$. Vamos escolher $y = 13$. O nosso problema é expresso pela seguinte função,

$$f(n) = 13^n \mod N. \tag{5.33}$$

Dada a complexidade do exemplo, facilmente calculamos manualmente o período da função, bastando para tal experimentar diferentes n's.

$$\begin{aligned} n = 1 &\Rightarrow 13 \mod 15 \equiv 13, \\ n = 2 &\Rightarrow 13^2 \mod 15 \equiv 4, \\ n = 3 &\Rightarrow 13^3 \mod 15 \equiv 7, \\ n = 4 &\Rightarrow 13^4 \mod 15 \equiv 1, \end{aligned}$$

de onde percebermos que r = 4. Vamos agora focar a nossa atenção no exemplo quântico. Para simplificar o circuito vamos definir t = L = 4. Assim

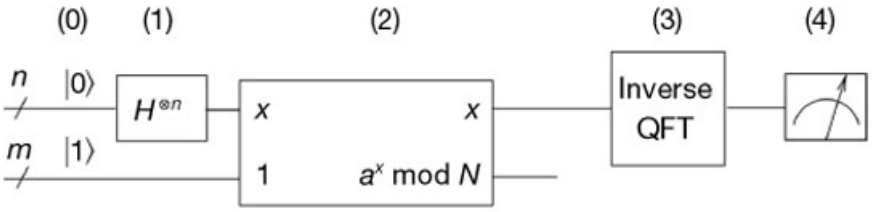


Figura 5.1: Circuito Quântico que implementa o algoritmo de Shor

$2^L \geq N$. Isso implica que estamos na posse de dois registos, cada um deles com 4 *qubits*.

Analisando o circuito apresentado na figura 5.1 da esquerda para a direita e começando por (1), obtemos,

$$(1) \equiv \frac{1}{\sqrt{16}} \sum_{k=0}^{15} |k\rangle |1\rangle^{\otimes n}, \tag{5.34}$$

onde para cada elemento $|k\rangle$ se verifica a seguinte analogia,

$$\begin{aligned} |0\rangle &\equiv |0000\rangle \\ |1\rangle &\equiv |0001\rangle \\ &\vdots \\ |15\rangle &\equiv |1111\rangle. \end{aligned} \tag{5.35}$$

É aplicado ao segundo registo a exponenciação modular. Recordando o cálculo do período efetuado no início da seção e estendendo o mesmo para $0 \leq n \leq 15$(superposição no primeiro registro), condensa-se na seguinte tabela o resulto do cálculo de $f(n)$.

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
f(n)	1	13	4	7	1	13	4	7	1	13	4	7	1	13	4	7

assim sendo, $|\psi_2\rangle$ é dado por,

$$(2) \equiv \frac{1}{\sqrt{16}} \sum_{k=0}^{15} |k\rangle \Big| x^k \mod 15 \Big\rangle \tag{5.36}$$

ou seja, neste momento estamos na posse do estado,

$$\begin{aligned} (2) \equiv \frac{1}{\sqrt{16}} \Big(& |0\rangle|1\rangle + |1\rangle|13\rangle + |2\rangle|4\rangle + |3\rangle|7\rangle + |4\rangle|1\rangle + |5\rangle|13\rangle + \\ & |6\rangle|4\rangle + |7\rangle|7\rangle + |8\rangle|1\rangle + |9\rangle|13\rangle + |10\rangle|4\rangle + \\ & |11\rangle|7\rangle + |12\rangle|1\rangle + |13\rangle|13\rangle + |14\rangle|4\rangle + |15\rangle|7\rangle \Big). \end{aligned} \tag{5.37}$$

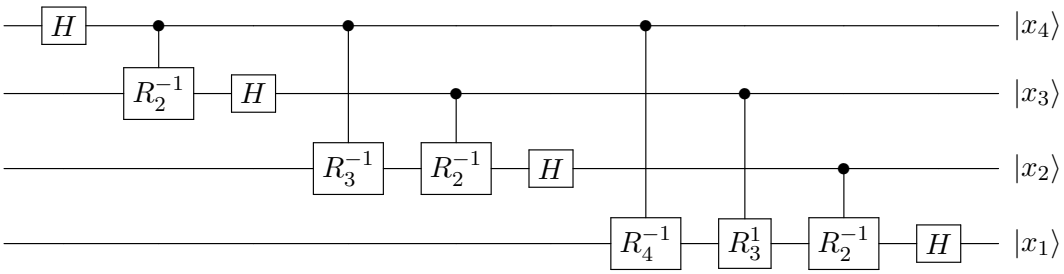


Figura 5.2: Circuito que implementa a TFQ_4^{-1}

Pelo princípio da *medição implicita*[4] medimos o segundo registro. Os valores deste estão contidos no conjunto, $\{|1\rangle, |13\rangle, |4\rangle, |7\rangle\}$, com probabilidade de ocorrência de 1/4 cada. Supondo que obtemos o valor 4, vamos selecionar do primeiro registro os estados que possuem período 4. São estes estados que servem de *input* a $TFQ^{\dagger}$, cujo circuito se apresenta na Figura 5.2. Podemos definir o estado $|\psi_3\rangle$ por,

$$(3) \equiv \sqrt{4} \times \frac{1}{\sqrt{16}} \Big(|2\rangle + |6\rangle + |10\rangle + |14\rangle \Big), \tag{5.38}$$

onde o fator $\sqrt{4}$ resulta da normalização do estado. Isto é, considere-se o seguinte exemplo,

$$|\psi\rangle \equiv \alpha|00\rangle + \alpha|01\rangle + \alpha|10\rangle + \alpha|11\rangle,$$

sujeito à seguinte condição de normalização, $\|4\alpha\|^2 = 1$. Ou seja, podemos multiplicar o estado $|\psi\rangle$ pelo respectivo valor de normalização $\sqrt{4}$, normalizando-o. Posto isto, o estado indicado por (3) vai ser sujeito a seguinte transformação,

$$|k\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u k}{16}} |u\rangle \tag{5.39}$$

cuja ação em cada elemento se encontra representada de seguida,

$$\begin{cases} |2\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.2}{16}} |u\rangle \\ + \\ |6\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.6}{16}} |u\rangle \\ + \\ |10\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.10}{16}} |u\rangle \\ + \\ |14\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.14}{16}} |u\rangle. \end{cases} \tag{5.40}$$

[4]Qualquer fio quântico por terminar, isto é, qubits que não foram medidos no fim do circuíto, podem assumir-se medidos

Condensando agora este resultado, podemos rescrever o estado de saída como se segue,

$$(4) \equiv \frac{\sqrt{4}}{\sqrt{16}} \sum_{u=0}^{15} |u\rangle \left\{ e^{2\pi i.2/16} + e^{2\pi i.6/16} + e^{2\pi i.10/16} + e^{2\pi i.14/16} \right\}, \quad (5.41)$$

que podemos facilmente calcular recorrendo ao SAGE(D.2). O resultado deste procedimento é um vector estado na forma $\sum_u \alpha|u\rangle$ com uma determinada distribuição de probabilidades em α. Isto é, no nosso exemplo ao calcularmos os 16 casos obtemos $P_0 = P_4 = P_8 = P_{12}$ com probabilidade de 1/4, com todas as restantes probabilidades iguais a zero. Desta forma o estado final será um dos elementos com a dita amplitude, respectivamente,

$$|\psi_{final}\rangle \in \{|0\rangle, |4\rangle, |8\rangle, |12\rangle\}. \quad (5.42)$$

Procuramos agora, cálcular uma aproximação ao período r na forma $|\psi_{final}\rangle \equiv s/r$. Por Nielsen and Chuang [2000], sabemos que $|\psi_{final}\rangle$ é um numero *racional*. Mais ainda, também sabemos a precisão que este possui (número de qubit's t). O próximo passo, é então calcular a fração mais próxima de $|\psi_{final}\rangle \equiv s/r$. O algoritmo das frações contínuas, permite-nos exatamente isto extraindo o **r**. Vamos demonstrar o mesmo, para cada um dos possíveis estados que obteríamos como *output* da medição do primeiro registo,

$$\frac{|0\rangle}{2^4} \to \frac{0}{16} \Rightarrow \text{sem informação - nova iteração com novo co-primo} \quad (5.43)$$

$$\frac{|4\rangle}{2^4} \to 0 + \frac{1}{4} \Rightarrow r = 4 \quad (5.44)$$

$$\frac{|8\rangle}{2^4} \to 0 + \frac{1}{2} \Rightarrow r = 2 \quad (5.45)$$

$$\frac{|12\rangle}{2^4} \to 0 + \frac{3}{4} \to 0 + \frac{1}{\frac{4}{3}} \to 0 + \frac{1}{1 + \frac{1}{3}} \to \frac{3}{4} \Rightarrow r = 4. \quad (5.46)$$

Entramos agora na última fase do algoritmo onde avaliamos r. Como constatamos possuímos dois resultados. Vamos analisar os dois. Relembrando o que foi dito no início do capítulo, na seção de *Teoria dos Números*, sobre o procedimento a seguir quando r é par, concluímos para $r = 2$ que,

$$13^{2/2} = 13 \neq -1 \mod 15 \quad (5.47)$$

Executando agora o algoritmo *Euclides*, obtemos

$$gdc(12, 15) = 3 \wedge gcd(14, 15) = 1 \to 15 \neq 3 \times 1 \to \text{falha!!}. \quad (5.48)$$

Por outro lado, agora considerando r = 4, verificamos,

$$13^{4/2} = 4 \neq 1 \mod 15 \quad (5.49)$$

e pelo algoritmo de *Euclides* obtemos,

$$gcd(13^2 + 1, 15) = 3 \wedge gcd(13^2 - 1, 15) = 5 \rightarrow 15 = 3 \times 5!. \tag{5.50}$$

Ou seja o algoritmo converge para a solução correta com 50% de probabilidade.

5.4 Sumário

Neste capítulo procurou-se analisar em detalhe o algoritmo de *Shor*. O mesmo utiliza além das propriedades quânticas já analisadas, conceitos de *teoria dos números que procuramos rever*. Nos mesmos, procuramos explicar a "truque" que o algoritmo usa, por forma a transformar o problema de fatorização, num problema que consegue ser implementado recorrendo a uma sub-rotina que faz uso da transformada de *Fourier* e cujo desempenho sabemos ser extraordinário. Posto isto, concretizamos o algoritmo com a resolução de um pequeno exemplo.
No capítulo que se segue, procuramos analisar concretamente o impacto que os algoritmos estudados por nós produzem numa família criptográfica muito dependente do problema da fatorização e logaritmo discreto. Iremos mesmo mais longe, mencionando outros algoritmos quânticos existentes e descrevendo o impacto destes, numa família criptográfica diferente daquela mais afetada pelo algoritmo de *Shor*. Além disto, procuramos ainda saber que soluções clássicas a criptografia atual oferece, que sejam eventualmente seguras na presença de um computador quântico. Abordaremos por fim, algoritmos quânticos, capazes de resolver problemas criptográficos atuais.

Capítulo 6

Impacto da Computação Quântica na Criptografia Atual

If computers that you build are quantum,
Then spies everywhere will all want 'em.
Our codes will all fail,
And they'll read your email,
Till we get crypto that's quantum,
and daunt 'em.

Jennifer and Peter Shor

Nesta seção do documento, procura-se descrever o impacto da computação quântica nos principais esquemas criptográficos em uso atualmente. Como tal, vamos começar por descrever os principais ou mais usados esquemas criptográficos. Nomeadamente, *criptografia simétrica* e *assimétrica*, descrevendo para cada uma em que medida, os algoritmos quânticos existentes comprometem a sua fiabilidade. Em particular, na criptografia assimétrica, apresentaremos o algoritmo provavelmente mais usado a nível mundial, o *RSA*, com recurso a uma experiência simplista do mesmo. Com esta abordagem, procuramos ilustrar claramente o ponto fulcral onde o algoritmo de fatorização proposto por *Peter Shor* e por nós analisado, intervém, permitindo ao atacante aceder à informação cifrada.

Em seguida e dado que praticamente comprometemos o standard criptográfico atual, veremos a reposta que a comunidade criptográfica propõem, de maneira a proteger-se deste novo modelo computacional, caso este algum dia veja a luz do dia e ganhe dimensões comerciais. Por último, apresentaremos dois algoritmos que se apresentam como soluções ao principal problema da criptografia simétrica - *distribuição de chave*. Com isto pretendemos en-

fatizar o contributo que pode advir da computação quântica, na criação de esquemas criptográficos mais seguros. As próximas páginas são adaptadas do trabalho de Zeng [2010].

6.1 Criptografia Simétrica

Este esquema criptográfico é provavelmente aquele cujo conceito é mais familiar para o leitor. Nesta configuração, ambos os participantes desejam trocar informação confidencial que se torne por isso indecifrável aos olhos de terceiros. Para tal, ambos chegam numa fase prévia da conversa, a acordo sobre uma chave secreta, que usam depois para cifrar e decifrar informação partilhada entre ambos. O acordo de chaves é precisamente o ponto mais sensível deste processo onde surgem normalmente os entraves à sua utilização. Além disso, os processos inerentes a este tipo de criptografia são mais leves, em comparação com os presentes na criptografia assimétrica.

Descreveremos agora muito sucintamente, uma classe de algoritmos conhecidos por *pesquisa quântica*. O seu propósito prende-se com a pesquisa em bases de dados não ordenada, atingindo-se reduções de pesquisa de nível quadrático. Os princípios básicos destes algoritmos foram descobertos por *Grover* e procuram resolver o seguinte problema: Dado um espaço de pesquisa de tamanho N, sem nenhum conhecimento prévio sobre a estrutura da informação existente, pretendemos encontrar nele um elemento que satisfaça uma propriedade conhecida.

Considerando uma base de dados indiferenciada, o custo de a pesquisar classicamente seria linear com o seu tamanho, $\Theta(N)$. Por outro lado, o *algoritmo de Grover* consegue efetuar a mesma tarefa em $\Theta(N^{\frac{1}{2}})$. Este algoritmo pode ser usado para pesquisar um determinado universo de chaves - *Ataque de força bruta* - ataque típico as cifras simétricas, encurtando o universo de chave a pesquisar, para cerca de metade. Se considerarmos a cifra simétrica *AES-128 bits*, este algoritmo precisaria de 2^{64} operações para a quebrar, contra as normais 2^{128}, que se acredita serem intratáveis para um computador corrente. Note-se que 2^{64} operações já não é considerado seguro pela comunidade criptográfica. Este é provavelmente o maior motivo pelo qual surgiu a cifra *AES-256 bits*.

Apesar de não se comparar com as reduções de caráter exponencial analisadas por nós, ao nível das operações necessárias para quebrar a cifra, esta classe de algoritmos é de grande interesse. Ao contrário dos algoritmos quânticos por nós estudados, cujo desempenho está diretamente relacionado com a dependência do algoritmo da transformada de *Fourier*. As suas potencialidades permitem ainda comprometer a segurança da cifra *DES* bem como

a de algumas funções de *hash*, nomeadamente, *MD5* e *SHA-1*, que também já foram consideradas inseguras clássicamente.

6.2 Criptografia Assimétrica

No sentido de tentar resolver o problema de distribuição de chaves imposto pela criptografia simétrica(distribuição da chave secreta através de um canal inseguro), surgiu no fim dos anos 70 um novo esquema criptográfico, conhecido por criptografia assimétrica ou de *chave pública*. Este tipo de criptografia, como foi mencionado, possui um maior peso computacional na codificação/descodificação da informação. Além disso, possui também uma 'filosofia" diferente de utilização. É neste tipo de criptografia que os computadores quânticos provocam um maior impacto[1], comprometendo a segurança de quase todos os esquemas criptográficos assimétricos. Nestes, cada interveniente tem um par de chaves respectivamente, *pública* e *privada*. A chave pública é passível de ser transmitida em aberto, podendo com ela qualquer pessoa cifrar informação que apenas o detentor da chave privada pode descodificar.
Este tipo de criptografia assenta em *convenções* profundamente enraizadas na dificuldade de resolver problemas matemáticos, no atual modelo computacional sem o conhecimento de todas as variáveis do problema. Como exemplo, e atendendo ao futuro caso prático por nós demonstrado, vamos considerar o problema de fatorização de dois números primos muito grandes (sensivelmente *1024 bits* cada):

1. Sejam P e Q, dois números primos de grande dimensão, e PQ o respectivo resultado da multiplicação.

2. Descartando P e Q, apenas estando na posse de PQ, queremos saber quais os fatores que lhe deram origem. isto é PQ.

Apesar do enunciado inocente, facilmente percebemos que o problema é complexo na medida em que a dimensão dos números escapa "às nossas mãos". Este problema matemático, está no cerne de um dos mais bem sucedidos esquemas criptográficos de chave pública, o *RSA*.
Voltando por agora à generalidade das cifras assimétricas e por forma a manter a coerência com a apresentação da cifras simétricas, vamos apresentar os principais pontos fortes e fracos desta.

- + Não é necessário os dois intervenientes chegarem a acordo sobre a chave secreta a usar.

- + Possibilita autenticação e o não repudio das mensagens.

- – É necessária a validação da autenticidade nas mensagens.

[1] Reduções de ordem exponencial

- – Custo acrescido na codificação/descodificação de informação.

Posto isto, avançaremos então para um exemplo concreto, o *RSA*, onde perceberemos como os algoritmos quânticos por nós estudados, comprometem a segurança desta cifra.

6.2.1 Exemplo - RSA

Criado em 1978 por três investigadores do MIT, nomeadamente, *Rivest*, *Shamir* e *Adlemen*, Rivest et al. [1978], o *RSA* veio a torna-se num dos principais esquemas criptográficos em uso no mundo inteiro. Como foi referido baseia-se num sistema de chaves pública/privada. Entre estas existem as seguintes dependências

1. Informação codificada com a chave pública pode apenas ser lida com a correspondente chave privada.

2. Informação codificada com a chave privada apenas pode ser lida pela chave publica.

3. Não existe uma relação óbvia entre as duas no sentido de, a partir da chave pública ser possível chegar à chave privada em tempo polinomial.

Devido ao grande custo computacional dos processos inerentes a este tipo de codificação/descodificação de informação, que analisaremos de seguida, este tipo de esquema é normalmente usado em simbiose com criptografia simétrica. A ideia é proteger por exemplo com uma cifra RSA, o acordo de chave, procedendo de seguida a "conversação" com a chave acordada, recorrendo a criptografia simétrica.
Vamos agora proceder a análise do algoritmo RSA. Para uma melhor legibilidade, vamos construir um pequeno exemplo de utilização. Com esta abordagem, procuramos clarificar o ponto onde algoritmo de *Shor* intervém, permitindo-nos quebrar a cifra.

1. Sejam **P** e **Q** dois números primos de grande dimensão (100 dígitos cada um). Para ilustrar o nosso exemplo, vamos escolher dos números mais pequenos, $\mathbf{P = 61}$ e $\mathbf{Q = 53}$.

2. Em seguida calculamos o produto $\mathbf{PQ = 3233}$. *Este valor é do conhecimento público.* Este valor será o coeficiente modular das operações a realizar.

3. Seja **E** a chave pública. Para escolha da mesma, o algoritmo garante, $\mathbf{E \leq PQ}$, **E é impar** e **não possui fatores comuns com (P-1)(Q-1)**.

4. Vamos admitir $\mathbf{E = 17}$. A *chave pública* é então o par **(PQ,E)**

5. Seja **D** a chave privada. O seu cálculo é dado pelo cálculo da *inversa multiplicativa* da chave pública **E**, módulo **PQ** isto é, **DE = 1 mod((P-1)(Q-1))**. **D = 2753**.

Neste momento estamos na posse do par chave pública, (**E**,**PQ**) e chave privada **D** que devemos guardar em absoluto sigilo. Vamos analisar agora as função de codificação/descodificação. Começando pela codificação, a sua função é dada por,

$$C(T) = T^E \mod PQ, \tag{6.1}$$

onde T é a informação a cifrar. Voltando ao nosso exemplo, e considerando T = 123, obtemos,

$$C(123) = 123^{17} \mod 3233 = 855 \tag{6.2}$$

Por outro lado a função de descodificação é dada por,

$$C^{-1}(V) = V^D \mod PQ, \tag{6.3}$$

no nosso exemplo, V é a informação cifrada, recebida pelo portador da chave privada. Substituindo a mesma na função de decifração, obtemos rapidamente a informação,

$$C^{-1}(855) = 855^{2753} \mod 3233 \equiv 123. \tag{6.4}$$

Devido as operações matemáticas envolvidas obter, **D**,**P**, **Q**, apenas com o conhecimento de **PQ** e **E** é considerado difícil. Relembrando a fórmula de calcular a chave privada, exposta no ponto 5 concretamente,

$$DE = 1 \mod (P-1)(Q-1), \tag{6.5}$$

percebemos que um atacante na posse de **E**, **PQ** e com recurso a um computador quântico, a correr o algoritmo de *Shor* por nós estudado, consegue fatorizar **PQ** nos seus dois fatores **P** e **Q**. Isto em tempo polinomial. O atacante consegue então deduzir a chave privada a partir da informação da chave publica.

De facto, a maior parte dos esquemas criptográficos baseados em criptografia de chave pública são sustentados por problemas matemáticos de difícil resolução, a não ser que se conheçam todas as variáveis inerentes ao mesmo. Este conhecimento é por vezes referido na literatura como *alçapão/porta traseira*[2] das funções de codificação/descodificação. Quem não as conhece, não consegue computar a função em tempo útil. Outro exemplo deste tipo de construção, são problemas baseados no *logaritmo discreto*[3]. Como exemplo temos a cifra *El Gamal*, o acordo de chaves *Diffie-Hellman*(Anexo C.1) ou as

[2] do inglês trap doors.

[3] `http://en.wikipedia.org/wiki/Discrete_logarithm`

assinaturas digitais. Ou seja, atendendo ao volume de informação mundial, que se encontra protegido por estas cifras (o que passa por todos os setores da sociedade moderna), a construção de um computador quântico com capacidade para correr estes algoritmos, provoca alguma apreensão à comunidade criptográfica bem como a sociedade.

Nos próximas seções deste capítulo, analisaremos tanto a resposta desta comunidade ao aparecimento deste algoritmos, como propostas da própria computação quântica no design de esquemas criptográficos mais seguros

6.3 Algoritmos Criptográficos Resistentes

Até ao momento, temos concentrado a maior parte do nosso esforço em compreender em que medida a construção de um computador quântico (capaz), vulnerabiliza os esquemas criptográficos atualmente existentes. No entanto, **acredita-se** que mesmo na presença de um computador quântico, existem soluções criptográficas atuais que perduram e como tal lhes sobrevivem.
Nesta seção procuraremos por isso dar a conhecer essa mesma criptografia resistente a computadores quânticos. Não pretendemos alongar-nos em demasia sobre o assunto. O mesmo mereceria um trabalho exclusivamente dedicado ao tópico. Ao invés, procuraremos referir o que existe, dando mais ênfase numa das soluções *Reticulados*[4], direcionando o leitor interessado para as diversas literaturas sobre o assunto. As próximas páginas baseiam-se no trabalho desenvolvido por Regev [2006] e Bernstein bem como, material audiovisual.[5]

Apresenta-se de seguida, as famílias de esquemas criptográficos que se acredita, serem resistentes a computadores quânticos.

- **Critptografia baseada em funções de hash** O exemplo clássico é a *Merkle's hash-tree public-key signature system (1979)*, cujos fundamentos advém da assinatura de mensagens únicas propostos por *Lamport* e *Diffie*.

- **Criptografia baseada em códigos.** A abordagem clássica advém de *McEliece's hidden-Goppa-code public-key encryption system (1978)*

- **Criptografia baseada em reticulados.** Esta abordagem é provavelmente a que mais entusiasma a comunidade criptográfica e provavelmente a mais promissora. Devido a isso dedicamos-lhe algumas linhas de seguida, por forma a tentar explicar em grosso modo em que é que ela consiste e qual as principais vantagens em relação a criptografia "standard"(baseada em fatorização). A título de curiosidade,

[4]Do inglês Lattices
[5]`http://www.youtube.com/watch?v=4ulHOV8iLls`

o exemplo que provavelmente atraiu maior interesse data de 1998[6] - *Hoffstein–Pipher–Silverman "NTRU"*.

- **Criptografia baseada em equações quadráticas multivariada.** Onde como exemplo temos *Patarin's "HFEv" - public-key-signature system* que data de 1996.

6.3.1 Reticulados

De uma forma geral um reticulado não é mais que um conjunto de pontos, num espaço com mais do que uma dimensão. Esse conjunto de pontos define-se da seguinte forma,

$$L = \{a_1 v_1 + \ldots + a_n v_n | a_i \quad \text{inteiro}\} \tag{6.6}$$

tal que, $v_1 \ldots v_n \in \mathbb{R}^n$. Mais ainda, os vectores $v_1 \ldots v_n$ são linearmente independentes. Resumidamente, "pegamos" no conjunto de vectores independentes em $\mathbb{R}^n$ e em todas as combinações inteiras desses pontos, para formar um espaço que se estende em n-direções. A Figura 6.1 proporciona-nos uma representação simplista de um reticulado em R^2, bem como redutora, visto que não se estende até ao infinito. Aos elementos $v_1, \ldots, v_n$ presentes na mesma, chamamos de *base* do reticulado. Outra representação muito usada para um reticulado é $\mathbf{L(B)}$, em que $\mathbf{B}$ é uma matriz $n \times n$, cujas colunas representam os vectores base $v_1 \ldots v_n$. A denominação $\mathbf{L(B)}$ representa então o reticulado construído a partir da base $\mathbf{B}$.
Um dos aspectos fundamentais para a criptografia é a capacidade de um reticulado possuir mais do que uma base. A intuição do porquê de isto acontecer será brevemente explicada, quando falarmos sobre os problemas sobre reticulados. No entanto, a título de curiosidade apresenta-se na Figura 6.1 duas bases para o mesmo reticulado. Neste caso a base que é mantida secreta é v_1^* e v_2^*, publicando-se v_1 e v_2.

6.3.2 O problema chave dos reticulados

O principal problema computacional associado aos reticulados é descobrir, o vector mais curto do reticulado, **SVP** - *Shortest vector problem.* Imaginemos que estamos na posse do reticulado $\mathbf{L(B)}$, para uma determinada base $\mathbf{B}$. O **SVP** deve produzir como output o vector *não nulo* mais pequeno nesse reticulado. Na realidade, em termos práticos não é bem este o problema que nos interessa, mas sim, uma aproximação ao **SVP**. Neste caso procuramos, **o vector *não nulo* cuja norma é maior do que um dado fator γ, que a norma do menor vector *não nulo* do reticulado**. Relembrando o que foi dito sobre as bases de um reticulado é mais ou menos intuitivo perceber que se publicarmos uma base com vectores muito grandes, tipicamente

[6] Não confundir com a data histórica onde apareceram os reticulados

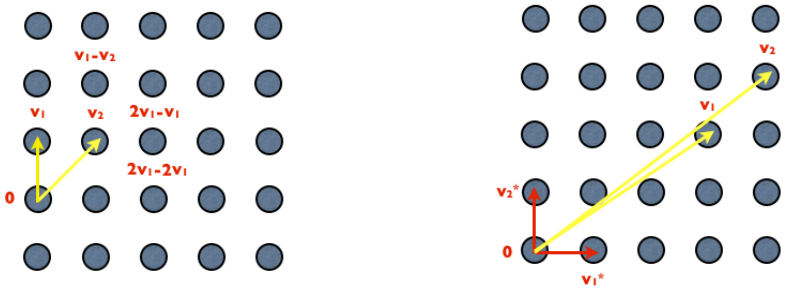


Figura 6.1: À esquerda temos um reticulado em R^2, com os vectores v_1 e v_2 a formar uma base. É possível observar algumas operações possíveis no reticulado, como o cancelamento de vectores $(-2v_1)$ por forma a obter os diferentes pontos no reticulado. A direita apresenta-se um Reticulado em R^2 onde é possível observar duas possíveis bases para o mesmo reticulado.

com várias ordens de grandeza em relação ao mais pequeno vector não nulo, descobrir esse vector é um problema complicado e moroso. Um dos mais conhecidos e eficientes algoritmos sobre reticulados, **LLL**-Lenstra [1982] consegue aproximar γ com um custo $2^{\theta(n)}$, onde n é a dimensão do reticulado. Existem algumas aplicações pertinentes deste problema, duas das mais conhecidas são as funções de sentido único propostas por Ajtai [1996] bem como, esquemas criptográficos de chave pública publicados por Ajtai and Dwork [1997]. Inicialmente, estas aplicações serviam fundamentalmente como provas de conceito, uma vez que o uso realista destes esquemas traduzia-se em chaves que podiam facilmente atingir os *Gigabyte's*. Apesar de não decorrermos sobre as abordagens, convidamos a leitor interessado a consultar diversos trabalhos que muito melhoram as soluções iniciais, tornando-as comparáveis em termos de rapidez por exemplo ao RSA em algumas configurações. Nomeadamente, para as funções de sentido único os trabalhos de Lyubashevsky and Micciancio [2006], Peikert and Rosen [2006] e Micciancio [2007]. Assim como, os seguintes trabalhos publicados sobre esquemas de chave publica baseados em reticulados, Regev [2003] e Ajtai [2005].

6.3.3 Vantagens sobre a criptografia standard

Procuramos agora condensar as principais vantagens do uso de esquemas criptográficos baseados em reticulados. Esta análise, será feita contrastando-os com aqueles que temos vindo a abordar no seguimento deste trabalho.

- Criptografia baseada em retículos

    1. **Segurança que pode ser "provada"** - É possível construir uma prova que relaciona a função criptográfica que estamos a implementar com a resolução de um problema dito difícil em reticulados. Isto funciona como uma redução da ação de "quebrar" uma função

criptográfica, na capacidade de resolver o dito problema. Esta prova, dá-nos indícios fortes se o nosso esquema criptográfico não tem falhas fundamentais. Por seu lado, a redução também implica que caso sejamos capazes de resolver o dito problema criptográfico associado ao reticulado, todas as aplicações desse problema em reticulados também são resolúveis - que são tipicamente problemas difíceis em reticulados. A própria prova dá-nos pistas dos parâmetros corretos para obtermos o melhor nível de segurança.

2. **Baseada em problemas difíceis dos reticulados** - Problemas estáveis e creditados na comunidade científica, já têm cerca de 30 anos no que toca a algoritmos computacionais, mas que se continua a acreditar serem difíceis.
3. **Resiste (ainda) a computadores Quânticos** - Para já ainda não foi descoberto nenhum algoritmo quântico que quebre estes algoritmos baseados em reticulados. Já se investiga há algum tempo e ainda não existem progressos nesta área.
4. **Computações simples** - normalmente as operações em reticulados passam por adições modulares... muitas.

- Criptografia "standard"

1. **Nem sempre se consegue provar** - Às vezes esta prova existe, mas baseia-se na *dificuldade do caso médio*. Considere-se por exemplo o esquemas criptográficos baseados em fatorização. Assumimos como difícil, a fatorização de números escolhidos com uma determinada distribuição. O problema é como escolher esta distribuição? Não devemos por exemplo, escolher números que possuam fatores pequenos (como números pares), mas talvez existam outros...
2. **Segurança baseada na dificuldade do caso médio** - Podemos quebrar uma determinada configuração/instância de um esquema criptográfico, por exemplo o RSA, mas isso não implica que conseguimos fatorizar todos os números.
3. **Baseados na dificuldade de *factorizar* ou no problema do logaritmo discreto**.
4. **Conhecem-se algoritmos quânticos que os quebram**
5. **Custo computacional** - Por exemplo para um "chip" de um cartão, realizar a exponenciação modular pode ser limitativo.

6.4 Algoritmos Quânticos Criptográficos

Nesta seção procuramos apresentar um conjunto de algoritmos criptográficos, baseados nos princípios de mecânica quântica que procuram resolver os problemas criptográficos atuais, cujas soluções clássicas dependem das limitações

computacionais do presente. No entanto e como foi explicado, na presença de um computador quântico a maior parte ficaria rapidamente obsoleta, comprometendo o funcionamento da sociedade como a conhecemos. Mais ainda, ao fazer depender os esquemas criptográficos futuros dos princípios da mecânica quântica, particularmente, *o princípio de Heisenberg*(Anexo B.4) e o *teorema da não clonagem*(Anexo B.5), conseguimos uma base de segurança muito mais sólida que a atual, uma vez que esta depende das leis da própria física, ao em vez do poder computacional presente e do atual modelo de computação.

Considerando como caso de estudo o problema do *acordo de chaves*, que é parte integrante da criptografia simétrica e cuja solução atual passa normalmente pela criptografia assimétrica, procuramos ilustrar como obter chaves secretas seguras por via da *distribuição de chaves quântica* (**QKD**). Em direção a esse objetivo, começaremos por expor o problema entre mãos, seguido da solução atual e as suas deficiências/problemas. Avançaremos depois para a descrição de 4 módulos ou fases, nomeadamente, *codificação quântica*, *transmissão quântica*, *deteção de espião* e *destilação de chave*, que constituem os algoritmos de **QKD**. É nestes, que a comunidade criptográfica deposita a sua "fé", que eliminariam os *dogmas* sobre as limitações atuais dos computadores, além de se precaverem e prevalecerem na eventualidade da construção de um computador quântico.

6.4.1 O Problema

Como sabemos o problema de acordo de chaves é inerente a criptografia simétrica. Resumidamente nesta, dois participantes procuram comunicar em segredo através de um meio não seguro, cifrando e decifrando as mensagens entre ambos, recorrendo a uma chave secreta. Para chegarem à mesma, os participantes recorrem a um procedimento de acordo de chaves, em que se acordam as seguintes questões:

1. Geração de chaves.
2. Distribuição de chaves.
3. Armazenamento de chaves.
4. Atualização de chaves.

Ao imaginarmos um cenário a escala global, uma entidade que recorresse a este procedimento com os seus clientes, percebemos rapidamente que se torna impraticável.

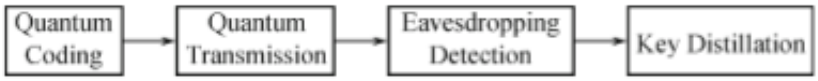


Figura 6.2: Diagrama das quatro fases básicas do acordo de chaves quântico

6.4.2 Esquemas criptográficos Quânticos

De uma forma geral e como já foi dito, um esquema **QKD** envolve 4 fases. Elas são, *codificação quântica*, *transmissão quântica*, *deteção de espião* e *destilação de chave* como é ilustrado na figura 6.2. Um dos objetivos desta seção é procurar transmitir a intuição sobre o funcionamento de cada uma destas fases. Este conhecimento será complementando na seção seguinte, onde apresentaremos dois algoritmos/esquemas QKD que as implementam, concretamente *BB84* e *BB92*.

Considere-se o par Alice e Bob.

1. **Codificação quântica**:

   - A Alice escolhe de forma aleatória *qubits* a partir de uma determinada linguagem **S**,

   $$S = \{s_i | i = 1, 2, \ldots, n\} \tag{6.7}$$

   - Com esses *qubits* procura codificar uma *string* R_c de bits. Por razões de segurança esta deve ser verdadeiramente aleatória e não pseudo-aleatória.

   $$R_c = \{r_i^c | i = 1, 2, \ldots, n\} \quad \text{t.q.} \quad r_i^c \in \{0, 1\}. \tag{6.8}$$

   - Considere-se a linguagem $S_{BB84} = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle$. A Alice escolhe de forma aleatória os símbolos da linguagem apresentada. Ao fazê-lo ela codifica uma qualquer *string* **R**, com uma determinada distribuição probabilística. Considere-se a seguinte **regra** aplicada pela Alice que é mantida **secreta** inclusive de Bob.

   $$0 \rightarrow |0\rangle \vee 0 \rightarrow |+\rangle \tag{6.9}$$
   $$1 \rightarrow |1\rangle \vee 1 \rightarrow |-\rangle \tag{6.10}$$

   - Neste momento a Alice está na posse de uma *string* $\mathbf{R} \in S_{BB84}$. Após esta fase ela envia a dita *string* para o Bob.

2. **Transmissão quântica**

- Cada elemento da *string* é normalmente transportado por fotões, componente indivisível da luz, ao longo de um canal de transmissão. Este último pode ser *fibra óptica* ou então aéreo.
- Podem existir dois tipos de transmissão, direta ou correlacionada. Na primeira, os *qubits* são diretamente transmitidos para o Bob. Este procedimento coincide com os mecanismos de transmissão que os algoritmos por nós apresentados na próxima seção utilizam. Na segunda, a transmissão não se assemelha à direta, recorrendo ao invés às propriedades do par *EPR*(Anexo B.3).
- Este processo é sustentando pelos princípios da *Teoria de Informação Quântica* Zeng [2010][pag.114], motivo pelo qual não discorreremos sobre ele.

3. **Deteção de Espião**

- Durante a fase de transmissão um espião, chamemos-lhe Eve, pode interceptar a mensagem. No entanto esta operação pode ser detectada e "combatida" recorrendo às leis da mecânica quântica, nomeadamente, *princípio de Heisenberg*(Anexo. B.4) e o *Teorema de Não Clonagem*(Anexo. B.5). Explicaremos melhor a sua ação na próxima seção.
- A ação do espião é julgada com base na taxa ou rácio de erros definida pelo algoritmo, resultante da transmissão dos *qubits*.
- A segurança de um esquema QKD, está relacionada com duas propriedades quânticas. A *não ortogonalidade* e *correlação*[7] que alguns estados quânticos permitem(Anexo B.3). Nos protocolos futuramente analisados por nós, consideraremos apenas a primeira.

4. **Destilação de Chave**

- Após terem terminado as fases em cima apresentadas, a Alice e o Bob possuem aquilo que é conhecido por uma chave em bruto. Esta não é chave final. Durante a comunicação, foram gerados erros em alguns bits provenientes da ação da Eve, bem como, possivelmente da própria comunicação. Mais ainda, a Eve pode ter conseguido obter alguns bits da chave em bruto.
- Para corrigir os erros citados, executa-se um processo de *Reconciliação* Zeng [2010][pag.117-125]
- Para aumentar a *privacidade* da chave final, executa-se um processo conhecido por *amplificação privada* Zeng [2010][pag.125-128], cujo objetivo é reduzir o tamanho da chave, a partir da chave em bruto, dai a palavra **destilar**. Podemos pensar nisto como, a execução de um função de *hash* específica, na chave em bruto.

[7] do inglês entanglement

- O processo de destilação de chave procura por isso corrigir estes dois casos. Os processos usados provém da *Teoria de Informação Clássica*, motivo pelo qual não dos debruçaremos com muito detalhe sobre eles.

Existem diferentes tipos de esquemas **QKD**. No entanto todos implementam as quatro fases aqui descritas. Mais ainda, neste trabalho apenas consideraremos os do tipo *standard*, concretamente o **BB84** e **B92**.

6.4.3 BB84

O protocolo BB84 foi o primeiro esquema **QKD** a ser apresentado. Concretamente, ele foi publicado em 1992 no *Journal of Cryptology* sendo proposto por *Bennett e Brassard*. À semelhança dos outros algoritmos apresentados neste trabalho, optou-se por fazer uma descrição do protocolo iterativa. Nesta, tentaremos dividir e ilustrar os 4 módulos descritos anteriormente e que fazem parte de qualquer esquema **QKD**.
Como sabemos a primeira fase do algoritmo traduz-se no processo de **codificação**. Para tal, a Alice prepara a seguinte linguagem **S**,

$$S_{BB84} = \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}. \tag{6.11}$$

nesta, os símbolos quânticos de adição e subtração traduzem os seguintes estados quânticos já nossos conhecidos.

$$|+\rangle \equiv \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{6.12}$$

$$|-\rangle \equiv \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{6.13}$$

Os símbolos da linguagem são equi-probabilísticos ($p = 1/4$). A Alice gera uma *string* completamente aleatória. Para efeitos do caso prático, vamos considerar a seguinte *string* R,

$$R = \{1100101101\}. \tag{6.14}$$

Do protocolo também sabemos que Alice tem a sua disposição uma **regra** que lhe permite codificar cada símbolo binário. Esta, utiliza as seguintes *bases* $\oplus$ e $\otimes$ para codificar cada bit, dando origem a diferentes *qubits*. Considere-se a seguinte regra, inerente ao protocolo BB84,

$$0 \Rightarrow \begin{cases} \oplus \rightarrow |0\rangle \\ \otimes \rightarrow |+\rangle \end{cases} \tag{6.15}$$

$$1 \Rightarrow \begin{cases} \oplus \rightarrow |1\rangle \\ \otimes \rightarrow |-\rangle \end{cases} \tag{6.16}$$

Analisando a mesma, percebemos que a Alice pode codificar os valores binários 0 e 1 de diferentes formas. Este processo de escolha é aleatório. A título de exemplo, considere-se a seguinte **regra** usada pela Alice.

$$Regra_{Alice} = \{\oplus, \otimes, \oplus, \otimes, \oplus, \oplus, \oplus, \oplus, \oplus, \oplus\}. \tag{6.17}$$

De momento ela mantêm-na **secreta**, não a partilhando com o Bob. A sua aplicação dá origem a seguinte *string* de *qubits*,

$$R_c = \{|1\rangle, |-\rangle, |0\rangle, |+\rangle, |1\rangle, |0\rangle, |1\rangle, |1\rangle, |0\rangle, |1\rangle\}, \tag{6.18}$$

Entramos agora na fase de **transmissão**, onde a *string* R_c é enviada ao Bob.

Cada símbolo quântico é transmitido ao Bob com intervalo Δt. Quando o Bob começa a recebe-los procura medí-los. Para tal, ele aleatoriamente seleciona uma base do 2-tuplo $\{\oplus, \otimes\}$. Dado o Bob não saber a regra que a Alice usou, isto implica que existe pelo menos 50% de hipótese de ele selecionar a base correta ao medir para cada bit. Vamos admitir que o Bob usou a seguinte regra,

$$Regra_{Bob} = \{\oplus, \otimes, \otimes, \otimes, \oplus, \oplus, \otimes, \oplus, \oplus, \otimes\} \tag{6.19}$$

tal implica que vão existir diferenças nos *bits* obtidos. Mas o Bob ainda não o sabe. Para o perceber, a Alice e o Bob recorrem a um canal de comunicação clássico. Neste, vão partilhar as regras usadas por ambos, descartando os *bits* que diferem. Entramos agora na fase de **detecção de espião**.

A Eve pode escutar o canal de transmissão. Graças ao *Teorema da Não Clonagem*(Anexo B.5), sabemos que a Eve não consegue copiar corretamente os *qubits* em transito, o que implica que ela só pode atacar o canal enquanto a comunicação decorre. Por outro lado, o princípio de *Heisenberg*(Anexo B.4), diz-nos que esta ação deturpa o canal. Colocando-nos no papel da Eve, percebemos que a única forma que ela tem de tentar adquirir os *qubits* em trânsito é medí-los. Para o fazer ela recorre aleatoriamente a medições nas bases $\oplus$ e $\otimes$. Isto dá-lhe 50% de hipóteses de acertar na base correta. Em qualquer dos casos, a Eve tem necessariamente de gerar o *qubit* que acabou de medir e que foi destruído, por forma a enviá-lo para Bob.
Após terem descartado os *bits* em que as bases utilizadas diferiam, a Alice e Bob precisam de fazer mais um teste por forma a despistar a presença da Eve. Imagine-se que a Alice utiliza a base $\oplus$, obtendo o *qubit* $|0\rangle$. A Eve por seu lado mede-o usando a base $\otimes$ obtendo o *qubit* $|-\rangle$ que envia para o Bob. O Bob usou a base $\oplus$, o que implica que aquando da comparação das bases este *bit* foi dado como válido. No entanto o Bob vai obter o bit 0 ou 1 com a sua medição e que pode deferir do da Alice. Por forma a detectar esta ação da Eve, eles aleatoriamente selecionam um conjunto de bits da string sacrificando-os e comparam-nos. Caso a comparação seja inferior ao limite

definido pelo algoritmo de 75%, os participantes abandonam o processo. Caso contrário o protocolo avança.

O protocolo agora, entra na fase de **destilação de chave**, cuja descrição já fizemos. Como foi dito e por entendermos que o procedimento desta se afasta dos princípios da mecânica quântica, não os abordaremos em mais detalhe.

6.4.4 B92

Este protocolo foi proposto de forma independente por *Bennett* em 1992. Criptograficamente, este protocolo é uma revisão do protocolo BB84. Vamos por isso concentrar-nos nas diferenças entre ambos.
Na fase de codificação a Alice prepara uma linguagem **S** dada por,

$$S_{B92} \equiv \{|\phi\rangle, |\psi\rangle\}, \tag{6.20}$$

onde $|\phi\rangle$ e $|\psi\rangle$ são dois *qubits* arbitrários e não ortogonais no espaço de *Hilbert*. isto é,

$$|\phi\rangle, |\psi\rangle \in \mathscr{H}, t.q, \parallel \langle\psi|\phi\rangle \parallel \neq 0. \tag{6.21}$$

A Alice gera uma *string* aleatória de *bits* codificando cada *bit* segundo a regra $0 \rightarrow |\phi\rangle$ e $1 \rightarrow |\psi\rangle$ ou vice-versa.

Existem diversas estratégias que o Bob pode usar para medir os resultados. Segundo Ekert et al. [1994], o Bob pode recorrer ao uso de **POVM**, definindo os seguintes operadores de medição,

$$E_1 \equiv \frac{I - |\phi\rangle\langle\phi|}{1 + \parallel \langle\phi|\psi\rangle \parallel} \tag{6.22}$$

$$E_2 \equiv \frac{I - |\psi\rangle\langle\psi|}{1 + \parallel \langle\phi|\psi\rangle \parallel} \tag{6.23}$$

$$E_3 \equiv I - E_1 - E_2. \tag{6.24}$$

Como sabemos, neste tipo de operadores de medição ou Bob mede corretamente o *qubit* transmitido pela Alice ou então não obtém nenhum valor, o *qubit* desaparece. Segundo *Ekert* a probabilidade de obtermos um resultado inconclusivo é dada por,

$$\parallel \langle\phi|\psi\rangle \parallel = \cos(2\theta), t.q \quad 0 < \theta < \pi/4. \tag{6.25}$$

Por seu lado, o Bob partilha com a Alice a posição dos *bits* nos quais ele conseguiu obter certeza ao medir. A partir desta chave temporária, os dois participantes vão selecionar um conjunto aleatórios de *bits* por forma a efetuarem testes de erros, a semelhança do protocolo BB84. O resto do protocolo segue por isso o mesmo caminho que o protocolo BB84.

6.5 Sumário

Neste capítulo, procuramos apresentar o impacto que a construção de um computador quântico pode provocar nos "standards" criptográficos atuais. Com o mesmo, procuramos salientar as vulnerabilidade dos esquemas atuais, exploradas pelos algoritmos quânticos existentes. Além disto, apontamos técnicas criptográficas existentes que se acreditem serem imunes a computadores quânticos.
No fim do capítulo discorremos sobre dois algoritmos quânticos. Estes, apresentam-se como soluções criptográficas ao problema da distribuição de chaves. Salientamos ainda que existem versões comerciais dos mesmos. Com esta breve revisão dos dois algoritmos, procuramos incutir que a criptografia quântica, não só necessariamente ameaça os esquemas criptográficos atuais, como pode trazer novas soluções criptográficas, resolvendo problemas com que os seus homólogos clássicos se debatem.

No próximo capítulo faremos uma revisão de todo o trabalho efetuado, salientando na perspectiva do autor, os aspectos mais positivos e menos bem conseguidos do mesmo.

Capítulo 7

Conclusão

To read our E-mail, how mean
of the spies and their quantum machine;
be comforted though,
they do not yet know
how to factorize twelve of fifteen.

Volker Strassen

Esta seção serve o propósito de comentar o trabalho desenvolvido numa perspectiva global. Nela procuramos transparecer no entendimento do autor, os aspectos mais positivos do trabalho desenvolvido, bem como apontar aspectos menos bem conseguidos do mesmo. Começando por estes últimos, a utilização da plataforma SAGE ficou aquém do esperado. A ideia inicial seria a de criar *workbook's* que permitissem o leitor acompanhar alguns dos conceitos estudados. Isto potenciaria a dinâmica do estudo. No entanto e pelo custo de aprendizagem que este trabalho significou para o autor o uso da plataforma SAGE foi algo descurado. No entanto, pequenos *scripts* de utilização foram desenvolvidos. A sua utilização permite animar, desde pequenos exemplos de calculo algébrico até algoritmos como o *two-level-unitaries* e parte do algoritmo de *Shor*.
Como aspecto mais positivo, considera-se o cumprimento do estudo inicialmente proposto. Concretamente, a aprendizagem de uma área completamente nova para o autor, a sua transmissão ao leitor, procurando fazê-lo de forma coloquial por forma a facilitar o seu entrosamento, recorrendo sempre que possível a pequenos exemplos. Além disso, procurou-se oferecer referencias suficientes para os leitores mais específicos e que procuram conhecimento nos aspectos mais levianamente analisados aqui. Analisaremos agora, os resultados mais esperados do trabalho, isto é, em que medida este novo modelo afeta a criptografia.

Durante este trabalho, a grande pergunta que procuramos responder era, em que medida a construção de um computador quântico em conjunção com

os algoritmos conhecidos podia afetar as técnicas criptográficas da atualidade. No fim do mesmo, percebemos que a resposta não era uniforme. Dividindo a ciência criptográfica em duas famílias fundamentais, respectivamente, simétrica e assimétrica, percebemos que os esquemas definidos na segunda são os mais afetados. Concretamente, problemas que sabemos terem custo exponencial, baixam abruptamente para problemas de cariz polinomial. Mais grave ainda é estes esquemas criptográficos serem aqueles que se encontram mais disseminados, pelos diferentes organismos da sociedade. Ou seja, na presença de um computador quântico, a maior parte dos segredos de estado, militares ou por exemplo transações bancárias pela internet, seriam facilmente comprometidas.
Por outro lado, no caso da criptografia simétrica este impacto é no máximo mais "suave". Concretamente, ele reduz o nível de segurança destas cifras para metade. No entanto e apesar do raciocínio tentador, de migrarmos o "standard" mundial de criptografia assimétrica para simétrica não é solução. A mesma tem problemas inerentes a sua filosofia que impossibilitam o seu uso massivo e exclusivo.

Todo o "impacto" demonstrado neste primeiro parágrafo, advém da construção de um computador quântico. Existe então razão para o pânico? Vamos dividir a resposta em duas partes. Em primeiro lugar, não se conhece a construção de um computador quântico capaz de correr os algoritmos quânticos existentes, com os recursos necessários para representar uma ameaça para as famílias criptográficas citadas. É verdade que já alguns foram construídos, nos quais se testou o mais famoso algoritmo quântico conhecido - *O Algoritmo de Shor*, mas o objetivo foi fatorizar um número que a maior parte de nós consegue fazer de cabeça. No entanto trabalhos recentes, atingiram a fatorização de um número da ordem das centenas. A reposta então a primeira parte da pergunta é que não é sensato descuidarmo-nos na construção de esquemas criptográficos. É verdade, que ainda não existe um computador quântico que quebre por exemplo uma cifra com características militares (tipicamente das mais seguras), ou pelo menos não é do domínio público que exista. No entanto, não é prudente ficarmos à espera que tal possa acontecer. Note-se que isto pode nunca acontecer, mas também pode surgir no mês seguinte.
A segunda parte da resposta encaixa no último ponto do parágrafo anterior. A comunidade criptográfica não ficou parada. Na realidade existem esquemas atuais que se **acredita** serem resistentes a um computador quântico. Esta confiança, advém do fato de estes esquemas não serem propriamente novos e de ainda não se ter descoberto nenhum algoritmo quântico para o efeito, apesar da investigação na área. Além disso, investigação tem sido feita no sentido de descobrir novos problemas, bem como provas de segurança, que nos permitem fazer assunções mais fortes sobre a segurança destes esquemas. A ideia é tentar garantir que o problema encontrado escapa ao domínio

quântico, o que nos permitira viver sem constante receio de "um algoritmo" que possa ser descoberto amanhã e que deite por terra o trabalho de hoje.

Por último também se gostaria de frisar que a construção de computadores quântico não se traduz apenas em ameaças para a criptografia. Na realidade existem também algoritmos quânticos criptográficos, inclusive à venda. Estes resolvem problemas com que os seus homólogos clássicos se debatem. Mas mais uma vez, ainda somos limitados pela tecnologia na sua aplicação a 100%.

Referências

Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996. **Cited** on page 94.

Miklós Ajtai. Representing hard lattices with o(n log n) bits. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, STOC '05, pages 94–103, New York, NY, USA, 2005. ACM. ISBN 1-58113-960-8. doi: 10.1145/1060590.1060604. URL `http://doi.acm.org/10.1145/1060590.1060604`. **Cited** on page 94.

Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, pages 284–293, New York, NY, USA, 1997. ACM. ISBN 0-89791-888-6. doi: 10.1145/258533.258604. URL `http://doi.acm.org/10.1145/258533.258604`. **Cited** on page 94.

D.J. Bernstein. Introduction to post-quantum cryptography. In *Post-Quantum Cryptography*, pages 1–14. Springer-Verlag. **Cited** on page 92.

Donny Cheung. Using generalized quantum fourier transforms in quantum phase estimation algorithms. Master's thesis, University of Waterloo, 2003. **Cited** on pages 58, 65, 67 and 70.

D. Deutsch and R. Jozsa. Rapid Solution of Problems by Quantum Computation. *Royal Society of London Proceedings Series A*, 439:553–558, December 1992. doi: 10.1098/rspa.1992.0167. **Cited** on page 62.

Artur Ekert and Richard Jozsa. Quantum computation and shor's factoring algorithm. *Rev. Mod. Phys.*, 68:733–753, Jul 1996. doi: 10.1103/RevModPhys.68.733. URL `http://link.aps.org/doi/10.1103/RevModPhys.68.733`. **Cited** on pages 76 and 80.

Artur K. Ekert, Bruno Huttner, G. Massimo Palma, and Asher Peres. Eavesdropping on quantum-cryptographical systems. *Phys. Rev. A*, 50:1047–1056, Aug 1994. doi: 10.1103/PhysRevA.50.1047. URL `http://link.aps.org/doi/10.1103/PhysRevA.50.1047`. **Cited** on page 101.

Ian Glendinning. The Bloch Sphere. Descição dos diferentes passos necessários a representação de um estado quântico na Bloch Sphere, February 2005. **Cited** on page 32.

Lenstra A.K. Lovász L. Lenstra, H.W. jr. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. URL `http://eudml.org/doc/182903`. **Cited** on page 94.

Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Ingo Wegener, Vladimiro Sassone, and Bart Preneel, editors, *Proceedings of the 33rd international colloquium on automata, languages and programming - ICALP 2006*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155, Venice, Italy, July 2006. Springer-Verlag. **Cited** on page 94.

Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. **Cited** on page 94.

M.A. Nielsen and I.L. Chuang. *Quantum computation and quantum information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2000. ISBN 9780521635035. URL `http://books.google.com/books?id=65FqEKQOfP8C`. **Cited** on pages 7, 16, 19, 20, 23, 25, 27, 40, 47, 50, 53, 57, 65, 70, 73, 77, 84 and 118.

Nicholas Ouellette. Quantum computation. Master's thesis, Swarthmore College, 2002. **Cited** on pages 51, 57 and 76.

Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *In TCC*, pages 145–166. Springer, 2006. **Cited** on page 94.

Oded Regev. New lattice based cryptographic constructions. *CoRR*, cs.CR/0309051, 2003. **Cited** on page 94.

Oded Regev. Lattice-based cryptography. In *Advances in cryptology (CRYPTO)*, pages 131–141, 2006. **Cited** on page 92.

R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978. ISSN 0001-0782. doi: 10.1145/359340.359342. URL `http://doi.acm.org/10.1145/359340.359342`. **Cited** on page 90.

Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. **Cited** on pages 75 and 81.

Guihua Zeng. *Quantum Private Communication*. Springer, Dordrecht, 2010. **Cited** on pages 88 and 98.

Apêndice A

Teoria dos Números

Nesta seção apresentamos alguns conceitos ou algoritmos utilizados no corpo da dissertação. Na exposição dos mesmos procura-se transmitir a componente prática destes, descartando-se as provas matemáticas dos variados conceitos aqui expostos.

A.1 Algoritmo de Euclides

Dados n_1 e n_2 inteiros, o algoritmo de *Euclides* é um método eficiente para calcular o maior divisor comum, $gcd(n_1, n_2)$. Seja $n_1 \geq n_2$. Começa-se por dividir n_2 por n_1, considerando r_1 como resto da sua divisão inteira,

$$n_1 = k_0 n_2 + r_1, r_1 < n_2. \tag{A.1}$$

Procedendo da mesma forma com n_2 e r_1,

$$n_2 = k_1 r_1 + r_2, r_2 < r_1. \tag{A.2}$$

Agora repetimos o mesmo passo para os dois r's

$$r_1 = k_2 r_2 + r_3, r_3 < r_2, \tag{A.3}$$
$$r_2 = k_3 r_3 + r_4, r_4 < r_3, \tag{A.4}$$

até o resto ser zero, o que acontece eventualmente, uma vez que o r está continuamente a diminuir,

$$r_{l-1} = k_l r_l + r_{l+1}, r_{l+1} < r_l, \tag{A.5}$$
$$r_l = k_{l+1} r_{l+1} + 0. \tag{A.6}$$

O maior divisor comum, $gcd(n_1, n_2)$ é então dado pelo último resto diferente de zero, isto é,

$$gcd(n_1, n_2) = r_{l+1}. \tag{A.7}$$
$$r_{l+1} = r_{l-1} - k_l r_l. \tag{A.8}$$

Se agora substituirmos a primeira equação na segunda, é possível exprimir $gcd(n_1, n_2)$ como uma combinação linear de n_1, n_2, dada por,

$$gcd(n_1, n_2) = a n_1 + b n_2, \tag{A.9}$$

onde a e b são inteiros que dependem dos $k_i's$. Se agora,

$$an_1 + bn_2 = 1 \tag{A.10}$$

isto é, n_1 e n_2 são co-primos, isto implica que

$$an_1 \equiv 1 \mod n_2, \tag{A.11}$$

o que se traduz em $a \equiv n_1^{-1} \mod n_2$. Logo afirma-se que números co-primos possuem sempre inversa multiplicativa módulo n_2. Vamos agora calcular, gcd(6825,1430)

$$\begin{aligned} 6825 &= 4 \times 1430 + 1105 \\ 1430 &= 1 \times 1105 + 325 \\ 1105 &= 3 \times 325 + 130 \\ 325 &= 2 \times 130 + 65 \\ 130 &= 2 \times 65 \end{aligned}$$

gdc(6825,1430) = 65.

A.2 Função Phi de Euler e a Ordem módulo N

A função de Phi *Euler*, $\varphi(N)$ é responsável para um determinado número N, retornar o número de inteiros menores que N que são co-primos de N. Então, se por exemplo, p for um número primo,

$$\varphi(p) = p - 1 \tag{A.12}$$

$$\varphi(mn) = \varphi(m)\varphi(n), \text{t.q, gcd(m,n)} = 1. \tag{A.13}$$

Esta função serve de base a muitos teoremas elegantes de teoria dos números. Por exemplo, o Teorema de *Euler*, afirma que,

$$a^{\varphi(N)} \equiv 1 \mod N \leftarrow gcd(a, N) = 1, \tag{A.14}$$

então se gcd(a,N) = 1, então implica que existe uma potência de **a** cuja aritmética modular N, é 1 ($1 \mod N$). A partir da menor potência de **a** que respeita esta propriedade, citamos a seguinte definição:

Teorema 5. *Se gcd(a,N) = 1. Então a ordem r de a* $\mod N$ *é a menor potência de* **a***, cujo resultado da aritmética modular é 1.*

Note-se que, se $gcd(a, N) \neq 1$, então nenhuma potência de $\mathbf{a} \equiv 1 \mod N$.

A.3 Frações Contínuas

A ideia do algoritmo das frações contínuas é a de descrever um número real usando apenas números inteiros, através de expressões na forma,

$$[a_0, \cdots, a_M] \equiv a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \cfrac{1}{a_M}}}}, \tag{A.15}$$

onde $a_0, \cdots, a_M$ são inteiros positivos. Para nós é conveniente permitir $a_0 = 0$. O algoritmo das frações continuas é um procedimento que permite calcular a expansão de um número real arbitrário. Isto é rapidamente compreendido com um exemplo. Considere-se o real 31/13. O primeiro passo do algoritmo é decompor 31/13 na sua componente inteira e fraccional, respectivamente,

$$\frac{31}{13} \equiv 2 + \frac{5}{13}. \tag{A.16}$$

Em seguida invertemos a parte fraccional, obtendo

$$\frac{31}{13} = 2 + \frac{1}{\frac{13}{5}}. \tag{A.17}$$

Estes dois passos são então aplicados recursivamente a fração 13/5, isto é

$$\frac{31}{13} \equiv 2 + \frac{1}{2 + \frac{3}{5}} \equiv 2 + \frac{1}{2 + \frac{1}{\frac{5}{3}}}. \tag{A.18}$$

de seguida separamos e invertemos 5/3,

$$\frac{31}{13} = 2 + \frac{1}{2 + \frac{1}{1 + \frac{2}{3}}} \equiv 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{\frac{3}{2}}}}. \tag{A.19}$$

Neste momento o processo de decomposição termina uma vez que,

$$\frac{3}{2} = 1 + \frac{1}{2} \tag{A.20}$$

pode ser escrito com um 1 no numerador sem necessidade de inverter. Nesse caso a representação final em termos das frações contínuas de 31/13 é

$$[2, 2, 1, 1, 2] \equiv 2 + \frac{1}{2 + \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}} \tag{A.21}$$

Analisando com atenção o resultado final, em particular os numeradores das diferentes frações continuas, é transparente que o algoritmo termina após um número finito de, divisões e inversões, para qualquer número racional, uma vez que estes (os numeradores) diminuem sistematicamente (31,5,3,2,1). Do algoritmo também sabemos que, se φ for um número real dado por $\varphi = s/r$, (tal que, s e r são inteiros de L bits), então o algoritmo computa a fração expandida usando $\Theta(L^3)$ operações. Cada uma destas operações usa $\Theta(L^2)$ operadores para realizar a aritmética.

A.4 Demonstração de $\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}|u_s\rangle = |1\rangle$.

A equação que pretendemos demonstrar, traduz-se na superposição de todos os vectores próprios de $|u_s\rangle$. Como tal, podemos definir os mesmos como,

$$|u_s\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1} e^{\frac{-2\pi isk}{r}}|x^k \mod N\rangle. \tag{A.22}$$

tal que, s é um numero inteiro no intervalo $0 \leq s \leq r-1$ e x, um valor aleatório menor que N mas que seja co-primo deste. Se agora substituirmos $|u_s\rangle$ na equação que pretendemos demonstrar obtemos,

$$\frac{1}{r}\sum_{s=0}^{r-1}\sum_{k=0}^{r-1} e^{\frac{-2\pi isk}{r}}|x^k \mod N\rangle \equiv \frac{1}{r}\sum_{k=0}^{r-1}\sum_{s=0}^{r-1} e^{\frac{-2\pi isk}{r}}|x^k \mod N\rangle. \tag{A.23}$$

Note-se a ação do somatório exterior no somatório interior,

$$\sum_{s=0}^{r-1} e^{\frac{-2\pi isk}{r}} = \begin{cases} r & \text{se } k = 0 \\ 0 & \text{se } k \neq 0 \end{cases} \tag{A.24}$$

onde no segundo caso, as r contribuições cancelam-se. Ou seja, se substituirmos este resultado na equação A.23, obtemos,

$$\frac{1}{r}\Big(r|1\rangle + \sum_{k=1}^{r-1}\sum_{s=0}^{r-1} e^{\frac{-2\pi isk}{r}}|x^k \mod N\rangle\Big) \equiv |1\rangle. \tag{A.25}$$

Apêndice B

Fundamentos da Mecânica Quântica

B.1 Procedimento Gram–Schmidt

Considere-se $|w_1\rangle, \ldots, |w_d\rangle$ como o conjunto de vectores que forma a base do espaço vectorial V e que é dotado da operação de produto interno. Graças ao método *Gram-Schimidt*, podemos construir uma base ortonormal, $|v_1\rangle, \ldots, |v_d\rangle$ para V, tal que,

$$|v_1\rangle \equiv \frac{|w_1\rangle}{\||w_1\rangle\|}. \tag{B.1}$$

Indutivamente, definimos agora de $1 \leq k \leq d-1$, os seguintes estados $|v_{k+1}\rangle$,

$$|v_{k+1}\rangle \equiv \frac{|w_{k+1}\rangle - \sum_{i=1}^{k} \langle v_i|w_{k+1}\rangle\, |v_i\rangle}{\||w_{k+1}\rangle - \sum_{i=1}^{k} \langle v_i|w_{k+1}\rangle\, |v_i\rangle\|}. \tag{B.2}$$

onde os vectores, $|v_1\rangle, \ldots, |v_d\rangle$, formam um conjunto ortonormal que também é uma base para V. A prova deste resultado pode ser consultada em Wikipedia Gram-Schmidt.

B.2 Comutador e Anti-Comutador

O comutador entre dois operadores A e B defini-se por

$$[A, B] \equiv AB - BA. \tag{B.3}$$

Se este for igual a zero, isto é, AB = BA, então diz-se que *A comuta com B*. Por sua vez o *anti-comutador* entre dois operadores expressa-se da seguinte forma,

$$\{A, B\} \equiv AB + BA \tag{B.4}$$

Afirma-se de forma semelhante que quando este é igual a zero então *A anti-comuta com B*. Para nós, estes operadores são importantes em dois aspectos. Primeiro, precisamos dele para compreender o princípio de incerteza de *Heisenberg*, e depois,

permite-nos derivar a propriedade de dois operadores hermitianos serem simultaneamente diagonizaveis. Considere-se A e B, dados por

$$A = \sum_i a_i |i\rangle\langle i|$$
$$B = \sum_i b_i |i\rangle\langle i|, \tag{B.5}$$

onde $|i\rangle$ é um qualquer conjunto ortonormal de vectores próprios para A e B.

Teorema 6 (Diagonalização simultanea)**.** *Suponhamos que A e B são dois operadores Hermitianos. Então [A,B] = 0 se e só se existir uma base ortonormal tal que os dois operadores são diagonizáveis em relação a essa base. Quando tal acontece, diz que A e B são simultaneamente diagonizáveis.*

Este teorema liga o conceito de comutador, que é simples de calcular, ao conceito de ser "diagonizável" que a *priori* é mais complicado. Como exemplo,

$$\begin{aligned}[X,Y] &= \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} - \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \\ &= 2i\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \\ &= 2iZ. \end{aligned} \tag{B.6}$$

Então X e Y não comutam. Intuitivamente podíamos conjeturar isto pelo facto de não existirem vectores próprios comuns entre os dois operadores. Por último apresentamos as relações de comutador/anti-comutador existentes entre as matrizes de *Pauli*

$$[X,Y] = 2iZ; \qquad [Y,Z] = 2iX; \qquad [Z,X] = 2iY. \tag{B.7}$$

B.3 Estado de Bell/Pares EPR

Um estado quântico fascinante e extremamente importante é o estado de *Bell* também conhecido por par *EPR*,

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{B.8}$$

Apesar do aspecto inócuo, este estado traz muitas surpresas. O estado de *Bell* quando medido apresenta então a seguinte propriedade muito interessante. Quando medimos o primeiro *qubit* podemos obter dois resultados:

- 0, com probabilidade $\frac{1}{2}$, alterando o estado $|\psi\rangle$ para o respectivo $|\psi'\rangle = |00\rangle$
- 1, com probabilidade $\frac{1}{2}$, deixando $|\psi'\rangle = |11\rangle$.

Ou seja, **a medição do segundo *qubit* dá sempre o mesmo resultado que a medição no primeiro *qubit*, motivo pelo qual dizemos que estas estão *correlacionadas***. Para além disso, outros tipos de operadores de medição podem ser aplicados ao primeiro ou ao segundo *qubit* mantendo-se no entanto a propriedade de correlação. Mais ainda, esta propriedade - **entrelaçadas** mantêm-se mesmo se os *qubits* forem separados . Isto esta na origem de um fenômeno quântico espetacular, conhecido por **teleportação quântica**.

B.4 O postulado da incerteza de *Heisenberg*

Este postulado é provavelmente o resultado mais conhecido da mecânica quântica. Considerando A e B dois operadores hermitianos, juntamente com o estado $|\psi\rangle$. Vamos supor que $\langle\psi|AB|y\rangle$ tem como o resultado o número complexo na forma $x + iy$, onde x e y são números reais. Note-se que,

$$\begin{aligned}\langle\psi|[A,B]|\psi\rangle &= 2iy \\ \langle\psi|\{A,B\}|y\rangle &= 2x.\end{aligned} \tag{B.9}$$

Tal implica que,

$$\|\langle\psi|[A,B]|\psi\rangle\|^2 + \|\langle\psi|\{A,B\}|\psi\rangle\|^2 = 4\|\langle\psi|AB|\psi\rangle\|^2, \tag{B.10}$$

então pela desigualdade de *Cauchy-Schwarz*

$$\|\langle\psi|AB|\psi\rangle\|^2 \leq \|\langle\psi|A^2|\psi\rangle\langle\psi|B^2|\psi\rangle \tag{B.11}$$

juntamente com o resultado expresso pela equação B.10, bem como, descartando os termos negativos, obtemos,

$$\|\langle\psi|[A,B]|\psi\rangle\|^2 \leq 4\langle\psi|A^2|\psi\rangle\langle\psi|B^2|\psi\rangle. \tag{B.12}$$

Sejam C e D dois observáveis, tal que, $A = C - \langle C\rangle$ e $B = D - \langle D\rangle$. Então substituindo isto na última equação obtemos o postulado de *Heisenberg*

$$\Delta(C)\Delta(D) \geq \|\frac{\langle\psi|[C,D]|\psi\rangle\|}{2}. \tag{B.13}$$

Este resultado gera normalmente a seguinte *incorreta* interpretação. A medição do observável C com uma determinada "precisão" dada por $\Delta(C)$ causa que o valor de D seja sujeito a uma "perturbação" quantificada por $\Delta(D)$, de tal forma que isto permite que a prévia equação, seja satisfeita a certos níveis.
Apesar de ser verdade que uma medição quântica causa uma perturbação ao estado a ser medido, este não é o conteúdo do postulado da incerteza. A correta interpretação do mesmo prende-se com, "se prepararmos um grande número de estados quânticos (idênticos), $|\psi\rangle$, e efetuarmos medições do observável C em alguns desses estados e D nos restantes, então o desvio padrão $\Delta(C)$ dos resultados C multiplicado pelo desvio padrão $\Delta(D)$ dos resultados D, satisfaz a desigualdade expressa pela equação B.13".

Exemplo:
Como exemplo da aplicação do principio de *Heisenberg*, considere-se os observáveis X e Y quando aplicados para medição do estado $|0\rangle$. Sabemos do capitulo B.2 que $[X,Y] = 2iZ$, então o princípio da incerteza diz-nos que,

$$\Delta(X)\Delta(Y) \geq \langle 0|Z|0\rangle = 1. \tag{B.14}$$

Logo $\Delta(X)$ e $\Delta(A) \geq 0$.

B.5 O Teorema da não-clonagem

Uma cópia quântica ou clonagem quântica é como o nome indica, o processo pelo qual se copia/duplica um estado arbitrário quântico, sendo que este processo não

deve alterar de nenhuma maneira o estado a copiar. Expressando esta operação recorrendo a notação de *Dirac* o que se pretenderá é o seguinte,

$$U|\psi\rangle_A|e\rangle_B = |\psi\rangle_A|\psi\rangle_B, \tag{B.15}$$

onde U representa o operador de cópia, $|\psi\rangle_A$ o estado a ser clonado, $|e\rangle_B$ a *acila* (estado auxiliar, da futura cópia) e $|\psi\rangle_B$ o estado final da copia que se encontra no mesmo estado que $|\psi\rangle_A$ estava. No entanto esta operação é proibida na maior parte dos casos, pelas leias da mecânica quântica, como se expressa no seguinte teorema,

Teorema 7 (Teorema da Não-Clonagem)**.** *Um estado quântico arbitrário não pode ser copiado exatamente sem perturbar/alterar o estado original a copiar.*

Demonstração. Suponhamos $|\psi\rangle_q$ como o estado a copiar. Então como demonstra a equação em cima, precisamos de um estado auxiliar $|e\rangle_B$, que tem de estar no mesmo *espaço* que o estado a copiar. Este estado tem que ser independente do estado a copiar cujo conteúdo desconhecemos. Como também sabemos o estado composto, $|\psi\rangle_q|e\rangle_B$ é formado pelo produto tensorial.

Podemos manipular este estado de duas maneiras. Ou realizamos uma observação, que *colapsa* o estado de forma irreversível num dos vectores próprios do observável, o que por razões óbvias não nos interessa. Ou então controlar a evolução do estado $|\psi\rangle_q$ através do operador U que é por restrição um operador unitário. Intuitivamente e recorrendo apenas a linearidade de U: se a cópia for permitida então de maneira geral,

$$U(2\psi) \otimes e = (2\psi) \otimes (2\psi). \tag{B.16}$$

linearmente e considerando o lado esquerdo da igualdade,

$$2U(\psi \otimes e) = 2(\psi \otimes \psi), \tag{B.17}$$

enquanto que no lado direito obteríamos,

$$4(\psi \otimes \psi). \tag{B.18}$$

Isto é uma contradição, logo o teorema da não clonagem é verdade.

Apêndice C

Algoritmos Auxiliares

Nesta seção dos anexos, procura-se apresentar algoritmos cuja importância nos assuntos discutidos na dissertação, relegamos para segundo plano.

C.1 Protocolo Diffie-Hellman

O método *Diffie-Hellman* serve o propósito de permitir chegar ao acordo, entre dois participantes, a uma chave secreta usada posteriormente para codificar/descodificar informação, enviada através de um meio inseguro. Vamos ilustrar o seu procedimento recorrendo aos nossos conhecidos agentes, Alice e Bob.
Em primeiro lugar, os participantes acordam dois números primos, chamemos-lhes **g** e **p**. O número **p** deve ser de grande dimensão(no mínimo 512 bits), enquanto que **g** é uma raiz-primitiva **módulo p**. Estes números podem ser tornados públicos. Agora cada participante, escolhe um número aleatório grande, que vai corresponder à sua chave privada, respectivamente, **a** (Alice) e **b** (Bob). De seguida a Alice calcula,

$$A = g^a \mod p \tag{C.1}$$

enviando **A** ao Bob. Enquanto que o Bob faz o mesmo procedimento, calculando,

$$B = g^b \mod p, \tag{C.2}$$

enviando **B** para a Alice.
A fase seguinte é onde se calcula a seguinte chave secreta, **K**,

$$K = g^{ab} \mod p \tag{C.3}$$

calculada individualmente pela Alice e pelo Bob da respectiva forma,

$$\text{Alice} \equiv K = B^a \mod p = \left(g^b\right)^a \mod p, \tag{C.4}$$

$$\text{Bob} \equiv K = A^b \mod p = \left(g^a\right)^b \mod p. \tag{C.5}$$

A partir deste ponto, a Alice o Bob podem usar K para codificar/descodificar informação entre ambos.

Como informação publica temos, **g**, **p**, C.1, C.2. Desta forma, alguém que esteja a espiar a conversa entre ambos, precisa de descobrir **K** para a conseguir descodificar. Tal só é possível, computando **a** a partir de C.1 bem como, **b** a partir de C.2. Este procedimento é conhecido por o *problema logaritmo discreto*, que é impraticável no atual modelo de computação clássico (para valores de p muito grandes). De grosso modo é sua segurança equivale a do problema *RSA*.

C.2 Single Qubit Decomposition

Demonstração. Nielsen and Chuang [2000][pag 175/176] Considerando a prova efetuada na pagina citada, pretendemos demonstrar a aplicação do teorema 3.39, concretamente,

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{C.6}$$

onde o operador U pode ser expresso como,

$$U = \begin{bmatrix} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & -e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} \end{bmatrix}. \tag{C.7}$$

Tirando partindo desta propriedade, demonstraremos a sua aplicação no operador *Hadamard*,

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{C.8}$$

Começamos por igualar cada elemento do operador **H**, ao seu homologo do operador genérico U respectivamente,

$$\begin{cases} e^{i(\alpha-\beta/2-\delta/2)} \cos\frac{\gamma}{2} & = \frac{1}{\sqrt{2}} \\ e^{i(\alpha+\beta/2+\delta/2)} \cos\frac{\gamma}{2} & = -\frac{1}{\sqrt{2}} \\ e^{i(\alpha+\beta/2-\delta/2)} \sin\frac{\gamma}{2} & = \frac{1}{\sqrt{2}} \\ -e^{i(\alpha-\beta/2+\delta/2)} \sin\frac{\gamma}{2} & = \frac{1}{\sqrt{2}} \end{cases} \equiv \begin{cases} \cos(\gamma/2) = & \frac{1}{\sqrt{2}} e^{-i(\alpha-\frac{\beta}{2}-\frac{\delta}{2})} \\ \cos(\gamma/2) = & -\frac{1}{\sqrt{2}} e^{-i(\alpha+\frac{\beta}{2}+\frac{\delta}{2})} \\ \sin(\gamma/2) = & \frac{1}{\sqrt{2}} e^{-i(\alpha+\frac{\beta}{2}-\frac{\delta}{2})} \\ \sin(\gamma/2) = & -\frac{1}{\sqrt{2}} e^{-i(\alpha-\frac{\beta}{2}+\frac{\delta}{2})} \end{cases} \tag{C.9}$$

igualando os termos iguais obtemos,

$$\begin{cases} ------ \\ \frac{1}{\sqrt{2}} e^{-i(\alpha-\frac{\beta}{2}-\frac{\delta}{2})} = -\frac{1}{\sqrt{2}} e^{-i(\alpha+\frac{\beta}{2}+\frac{\delta}{2})} \\ ------ \\ \frac{1}{\sqrt{2}} e^{-i(\alpha+\frac{\beta}{2}-\frac{\delta}{2})} = -\frac{1}{\sqrt{2}} e^{-i(\alpha-\frac{\beta}{2}+\frac{\delta}{2})} \end{cases} \equiv \begin{cases} ------ \\ e^{i(-\alpha+\frac{\beta}{2}+\frac{\delta}{2})} = -e^{i(-\alpha-\frac{\beta}{2}-\frac{\delta}{2})} \\ ------ \\ e^{i(-\alpha-\frac{\beta}{2}+\frac{\delta}{2})} = -e^{i(-\alpha+\frac{\beta}{2}-\frac{\delta}{2})} \end{cases} \tag{C.10}$$

de seguida, aplicou-se a formula de *Euler*,

$$\begin{cases} ------ \\ \cos(-\alpha+\frac{\beta}{2}+\frac{\delta}{2}) + i\sin(-\alpha+\frac{\beta}{2}+\frac{\delta}{2} = -\cos(-\alpha-\frac{\beta}{2}-\frac{\delta}{2}) - i\sin(-\alpha-\frac{\beta}{2}-\frac{\delta}{2}) \\ ------ \\ \cos(-\alpha-\frac{\beta}{2}+\frac{\delta}{2}) + i\sin(-\alpha-\frac{\beta}{2}+\frac{\delta}{2} = -\cos(-\alpha+\frac{\beta}{2}-\frac{\delta}{2}) - i\sin(-\alpha+\frac{\beta}{2}-\frac{\delta}{2}) \end{cases} \tag{C.11}$$

rescrevendo-se de seguida esta expressão recorrendo ao **cis**,

$$\begin{cases} ------ \\ \cos(-\alpha+\frac{\beta}{2}+\frac{\delta}{2}) + i\sin(-\alpha+\frac{\beta}{2}+\frac{\delta}{2}) = -1 \;\; cis(-\alpha-\frac{\beta}{2}-\frac{\delta}{2}) \\ ------ \\ \cos(-\alpha-\frac{\beta}{2}+\frac{\delta}{2}) + i\sin(-\alpha-\frac{\beta}{2}+\frac{\delta}{2}) = -1 \;\; cis(-\alpha+\frac{\beta}{2}-\frac{\delta}{2}) \end{cases} \tag{C.12}$$

sabemos que

$$cis(\pi) = \cos(\pi) + i\sin(\pi) \equiv -1 + 0 = -1 \tag{C.13}$$

então podemos rescrever a expressão de cima como,

$$\begin{cases} ------ \\ cis(-\alpha + \frac{\beta}{2} + \frac{\delta}{2}) = \ \ cis(\pi)cis(-\alpha - \frac{\beta}{2} - \frac{\delta}{2}) \\ ------ \\ cis(-\alpha - \frac{\beta}{2} + \frac{\delta}{2}) = \ \ cis(\pi)cis(-\alpha + \frac{\beta}{2} - \frac{\delta}{2}) \end{cases}$$
$$\equiv$$
$$\begin{cases} ------ \\ cis(-\alpha + \frac{\beta}{2} + \frac{\delta}{2}) = \ \ cis(\pi - \alpha - \frac{\beta}{2} - \frac{\delta}{2}) \\ ------ \\ cis(-\alpha - \frac{\beta}{2} + \frac{\delta}{2}) = \ \ cis(\pi - \alpha + \frac{\beta}{2} - \frac{\delta}{2}) \end{cases} \tag{C.14}$$

percebemos neste momento que podemos cancelar alguns dos termos, obtendo,

$$\begin{cases} ------ \\ \frac{\beta}{2} + \frac{\delta}{2} = \pi - \frac{\beta}{2} - \frac{\delta}{2}) \\ ------ \\ -\frac{\beta}{2} + \frac{\delta}{2} = \pi + \frac{\beta}{2} - \frac{\delta}{2}) \end{cases} \equiv \begin{cases} ------ \\ \beta + \delta = \pi \\ ------ \\ -\beta + \delta = \pi \end{cases} \equiv \begin{cases} ------ \\ \beta = \pi - \delta \\ ------ \\ (-\pi + \delta) + \delta = \pi \end{cases} \tag{C.15}$$

$$\begin{cases} ------ \\ \beta = 0 \\ ------ \\ 2\delta = 2\pi \rightarrow \delta = \pi \end{cases} \tag{C.16}$$

Agora na posse de $\beta = 0$ e $\delta = \pi$, basta-nos substituir estes valores nas duas equações que ficaram pendentes, para retirar-mos os valores de α e γ. Respectivamente,

$$\begin{cases} \cos(\frac{\gamma}{2}) = \frac{1}{\sqrt{2}} e^{-i(\alpha - 0 - \frac{\pi}{2})} \\ \sin(\frac{\gamma}{2}) = \frac{1}{\sqrt{2}} e^{-i(\alpha - \frac{\pi}{2})} \end{cases} \tag{C.17}$$

aplicando novamente a formula de *Euler* obtemos,

$$\begin{cases} \cos(\frac{\gamma}{2}) = \frac{1}{\sqrt{2}} \Big(\cos(-\alpha + \frac{\pi}{2}) + i\sin(-\alpha + \frac{\pi}{2}) \Big) \\ \sin(\frac{\gamma}{2}) = \frac{1}{\sqrt{2}} \Big(\cos(-\alpha + \frac{\pi}{2}) + i\sin(-\alpha + \frac{\pi}{2}) \Big) \end{cases} \tag{C.18}$$

Em seguida vamos tirar partido do facto de,

$$\cos(\frac{x}{2})^2 + \sin(\frac{x}{2})^2 = 1 \tag{C.19}$$

rescrevendo a prévia equação. Isto traduz-se em,

$$\left(\frac{1}{\sqrt{2}} \cos(-\alpha + \frac{\pi}{2}) + i\sin(-\alpha + \frac{\pi}{2}) \right)^2 + \left(\frac{1}{\sqrt{2}} \cos(-\alpha + \frac{\pi}{2}) + i\sin(-\alpha + \frac{\pi}{2}) \right)^2 = 1. \tag{C.20}$$

que podemos simplificar para,

$$\frac{\cos(-\alpha+\frac{\pi}{2})^2+(i\sin(-\alpha+\frac{\pi}{2}))^2+\cos(-\alpha+\frac{\pi}{2})^2+(i\sin(-\alpha+\frac{\pi}{2}))^2}{2}=1. \quad \text{(C.21)}$$

Vamos agora tirar partido das seguintes propriedades,

$$\sin^2(x)=\frac{1}{2}-\frac{1}{2}\cos(2x) \quad \text{(C.22)}$$

$$\cos^2(x)=\frac{1}{2}+\frac{1}{2}\cos(2x). \quad \text{(C.23)}$$

Se agora as aplicarmos à nossa equação obtemos,

$$\begin{aligned}&\frac{(\frac{1}{2}+\frac{1}{2}\cos(-2\alpha+\frac{2\pi}{2}))-(\frac{1}{2}-\frac{1}{2}\cos(-2\alpha+\pi))}{2}\\ &+\\ &\frac{(\frac{1}{2}+\frac{1}{2}\cos(-2\alpha+\pi))-(\frac{1}{2}-\frac{1}{2}\cos(-2\alpha+\pi))}{2}\end{aligned} \quad \text{(C.24)}$$

que podemos simplificar para,

$$\frac{1}{2}\cos(-2\alpha+\pi)+\frac{1}{2}\cos(-2\alpha+\pi)+\frac{1}{2}\cos(-2\alpha+\pi)+\frac{1}{2}\cos(-2\alpha+\pi)=2. \quad \text{(C.25)}$$

somando agora os diferentes cosenos,

$$2\cos(-2\alpha+\pi)=2\rightarrow\cos(-2\alpha+\pi)=1\Rightarrow\alpha=\pi. \quad \text{(C.26)}$$

Se agora substituirmos numa das quaisquer equações, facilmente retiramos γ. Concretamente, o mesmo têm o valor $-\frac{\pi}{2}$. □

Apêndice D

Scripts SAGE

Nesta seção do documento, apresenta-se os *scripts* desenvolvidos em SAGE. Os mesmos, foram desenvolvidos por forma a auxiliar bem como animar, os respectivos conceitos apresentados. Por questões de tamanho de página, não se apresenta sempre o *ouput* dos mesmos. No entanto e especialmente nesse caso, o output deste pode ser consultado na respectiva seção da dissertação. Note-se, que os resultados apresentados na dita seção sofreram arredondamentos por forma a simplificar os cálculos.
Na página seguinte apresenta-se então os ditos *scripts*.

D.1 Two Level Unitaries


```
%hide
```

Algoritmo **TLU**,

- **input** Matriz quadrada (Operador) de dimensão $n > 2$
- O algoritmo decompoem o input num conjunto de matrizes. Sendo **n** a dimensão da matriz de entrada então teremos **k** matrizes dadas por

$$k \leq \frac{n(n-1)}{2}$$

- **output** Um dicionario. Para aceder a cada entrada realizar d[indice], tal que, $n - 2 \leq indice \leq 1$.

```
def TLU(x):
    dictionary = {}
    U = x
    aux = 0
    if is_Matrix(U) and is_square(U):
        for column in range(U.ncols()-2):
            for line in [column+1..U.ncols()-1]:
                id = identity_matrix(CC,U.ncols())
                if x[line,column] != 0:
                    id[column,column]= U[column,column].conjugate()/ \
                                       (sqrt(abs(U[column][column])^2 + abs(U[line][column])^2))
                    id[line,column]=U[line,column]/ \
                                       (sqrt( abs(U[column][column])^2 + abs(U[line][column])^2 ) )
                    id[column,line]=U[line,column].conjugate()/ \
                                      (sqrt( abs(U[column][column])^2 + abs(U[line][column])^2 ) )
                    id[line,line]= (-1*U[column,column])/ \
                                       (sqrt( abs(U[column][column])^2 + abs(U[line][column])^2 ))
                dictionary[aux+1]= id
                aux=aux+1
                U = id*U
    else:
        raise TypeError("The argument must be a square matrix")
    return dictionary
```

```
%hide
```

Exemplo da aplicação do algoritmo a seguinte matriz,

$$F \equiv \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

O algoritmo vai produzir um **dicionario** com 5 entradas. Nelas estão contidas as matrizes $F_1F_2F_3F_4F_5$. Sabemos que a decomposição resulta em 6 matrizes neste caso (4*(4-1)/2). Como o procedimento para calcular F_6 é trivial, apresenta-se em V_6 o mesmo, seguido da verificação que o procedimento funciona.

```
f = matrix(CC,[[1/2,1/2,1/2,1/2],[1/2,1/2*I,-1/2,-1/2*I],[1/2,-1/2,1/2,-1/2],[1/2,-1/2*I,-1/2,1/2*I]])
d = TLU(f)
show(d[1])
show(d[2])
show(d[3])
show(d[4])
show(d[5])
aux = (d[5]*d[4]*d[3]*d[2]*d[1] * test)
show(teste)
v6 = matrix(CC,[[1,0,0,0],[0,1,0,0],[0,0,0.7071,-0.7071],[0,0,-0.7071*i, -0.7071*i]])
show(d[1].conjugate_transpose()*d[2].conjugate_transpose()*d[3].conjugate_transpose()*d[4].conjugate_transpose()*
.conjugate_transpose())
```

D.2 Transformada de Fourrier Quântica - Exemplo Shor, factorização do 15


```
%hide
```

Script python que aplica a seguinte transformada de *Fourier*.

$$|k\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u k}{16}} |u\rangle \tag{1}$$

1. **vectorofinput** - Contém os estados cujo período é igual ao obtido quando aplicado o princípio da medição implícita. No exemplo, para o período 4, obteve-se os estados $|2\rangle, |6\rangle, |10\rangle, |14\rangle$.
2. **numbertofactor** - parâmetro com o numero a fatorizar (15), para efeitos do controlo do somatório.
3. **numberofbits** - parâmetro com o numero de *bits* usados (4). O mesmo permite-nos normalizar cada elemento do vector estado. Esta ação é visível na variável **aux**.
4. Em suma, podemos ver a sua ação em cada base como,

$$\begin{cases} |2\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.2}{16}} |u\rangle \\ + \\ |6\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.6}{16}} |u\rangle \\ + \\ |10\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.10}{16}} |u\rangle \\ + \\ |14\rangle \rightarrow \frac{1}{\sqrt{16}} \sum_{u=0}^{15} e^{\frac{2\pi i u.14}{16}} |u\rangle . \end{cases} \tag{2}$$

5. A transformada retorna um **dicionario**.

```
def QFT(vectorofinput,numbertofactor,numberofbits):

    dictionary = {}

    for b in vectorofinput:
        aux = sqrt(len(vectorofinput)/(2^numberofbits))
        for a in range (numbertofactor+1):
            if (a not in dictionary):
                dictionary[a] = aux*(1/sqrt(2^numberofbits))* \
                                e^(2*pi*i*((b*a)/(2^numberofbits)))
            else:
                dictionary[a] +=aux*(1/sqrt(2^numberofbits))* \
                                e^(2*pi*i*((b*a)/(2^numberofbits)))
    return dictionary

d = QFT([2,6,10,14],15,4)
aux = d[4]
aux.simplify_full().norm()
```

$\frac{1}{4}$

D.3 Animações na Bloch Sphere

Nesta seção do documento, apresentamos algumas animações sobre estados quânticos. As mesmas recorrem à *framework* - *qutip*(http://code.google.com/p/qutip). Os *scritps* apresentados são em código *python*, e isto permite-nos configurar o ambiente SAGE[1] para correr os mesmos.

D.3.1 Operador Hadamard - H

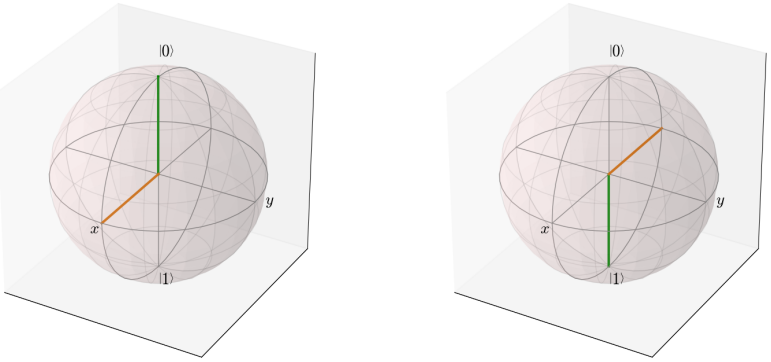


Figura D.1: À esquerda temos representada a ação do operador H no estado $|0\rangle$. Por sua vez à direita, temos o mesmo operador mas agora a atuar no estado $|1\rangle$. Note-se que a cor laranja esta associada ao estado pós-rotação, enquanto que a verde ao estado inicial

Apresenta-se de seguida o código *python* que deu origem ás animações. Note-se que ao invés de declarar duas esferas (uma para cada imagem) condensou-se as duas animações numa única esfera,

Phyton.

$$
\begin{aligned}
&>>> v1 = basis(2,0)\#|0\rangle \\
&>>> v2 = basis(2,1)\#|1\rangle \\
&>>> b = Bloch()\#\text{declaração da esfera de Bloch} \\
&>>> b.add\_states(v1) \\
&>>> b.add\_states(v2) \\
&>>> b.add\_states(snot() * v1)\#\text{adiciona-se o estado resultante de} H|0\rangle \\
&>>> b.add\_states(snot() * v2)\#\text{adiciona-se o estado resultante de} H|1\rangle \\
&>>> b.show()\#\text{consultar a esfera}
\end{aligned} \tag{D.1}
$$

□

[1] configuração não é apresentada

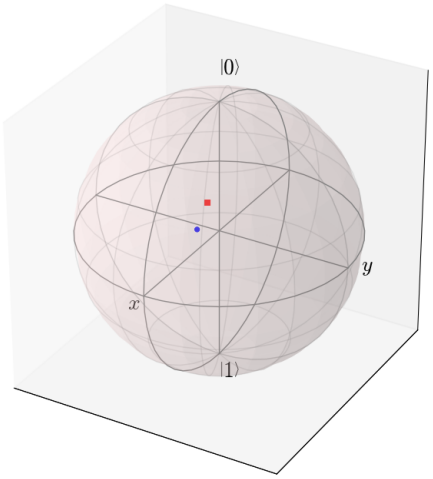


Figura D.2: Representação na esfera de Bloch do operador H a atuar no estado $|\psi\rangle$ com $\alpha = 0.5$ e $\beta = 0.3$. A cor azul, temos o estado inicial e a vermelho a transformação

Na figura D.2 expõe-se a ação do operador H, num estado quântico genérico, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$.

Phyton.

$>>> v1 = basis(2,0) * 0.5 \# 0.5|0\rangle$

$>>> v2 = basis(2,1) * 0.3 \# |1\rangle$

$>>> state = v1 + v2$

$>>> b = Bloch()$

$>>> b.add_states(state, "point") \#$escolhe-se representar o estado como um ponto

$>>> b.add\_states(snot() * state, "point")$

$>>> b.show()$ (D.2)

□

D.3.2 Operador S - Fase

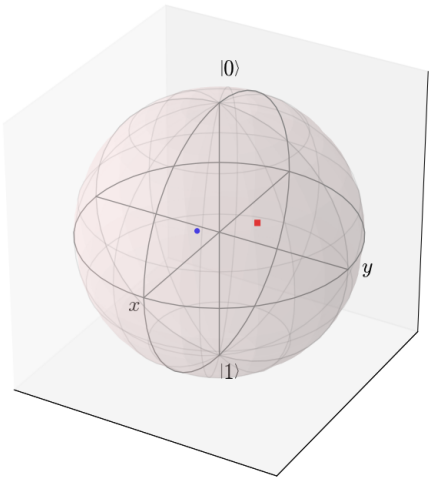


Figura D.3: Representação na esfera de Bloch do operador S, a atuar no estado $|\psi\rangle = 0.5|0\rangle + 0.3|1\rangle$. A cor azul, temos o estado inicial e a vermelho a referida transformação

Phyton.

$>>> v1 = basis(2,0) * 0.5$
$>>> v2 = basis(2,1) * 0.3$
$>>> state = v1 + v2$
$>>> b = Bloch()$
$>>> b.add\_states(state, "point")$
$>>> b.add_states(phasegate(pi/2) * state, "point")$aplicação do operador S
$>>> b.show()$ (D.3)

□