Fundamentos da Computação Quântica

3 de Setembro de 2019

Aluno: Bruno Vieira Ramos Silva (RA: 11201811306)[1]

Orientador: Nelson José Rodrigues Faustino (SIAPE: 2286843)[2]

Conteúdo

[1]email: bruno.vieira@aluno.ufabc.edu.br
[2]email: nelson.faustino@ufabc.edu.br

1 Prefácio

Este trabalho de iniciação científica teve como principal objetivo a realização de um estudo introdutório dos fundamentos da teoria de computação quântica, com especial enfoque nos algoritmos que vieram a popularizar esta área: o algoritmo de Deutsch-Jozsa (cf. [5]), o algoritmo de Grover (cf. [10]) e o algoritmo de Shor (cf. [26]).

A execução deste projeto ambicioso e ao mesmo tempo complexo, tinha tudo para dar errado. A começar pelo simples fato que este requeria, por parte do aluno, um domínio abrangente de Álgebra Linear e Física Quântica. Convém salientar que, ao contrário da disciplina de Física Quântica (BCK0103-15) que o aluno Bruno Silva teve possibilidade de cursar ao longo da execução do projeto, a disciplina de Álgebra Linear (BC-1425) não faz parte do plano curricular do Bacharelado em Ciência e Tecnologia (BC&T).

Neste sentido, os primeiros quatro meses do plano de trabalhos foram dedicados ao estudo destes conceitos. O capítulo 3 é meramente introdutório, e teve como principal objetivo o entendimento por parte do aluno de algumas partes do artigo de Feynman (cf. [7]), em particular a seção 3. Simulating Probability. O estudo de alguns conceitos basilares de álgebra linear foram fundamentais para entender propriedades intrínsecas a operadores Hermitianos que o aluno refere no final do capítulo 3, e no início do capítulo 4 onde o aluno introduziu os conceitos de **Quantum bit** e de **Produto Tensorial**.

Infelizmente, e por questões de tempo, não foi possível ir muito além do idealizado inicialmente, que incluía simulações computacionais usando a nuvem[3] IBM Q (ainda em fase beta) assim como o estudo da ordem de complexidade de alguns dos algoritmos. De qualquer das formas, como orientador congratulo-me de o aluno Bruno Silva ter desenvolvido grande parte do material que aparece neste relatório de forma independente e autónoma, incluíndo a edição do presente relatório no formato LaTeX.

Olhando agora pelo espelho retrovisor, posso afirmar com base no recente livro de *de Lima Marquezino et al* (2019) (cf. [4]) que o tema de computação quântica será de importância estratégica para o desenvolvimento científico e tecnológico do Brasil na próxima década. Pelo que tem vindo a ser publicado, de forma contínua na revista de divulgação científica online *Quanta Magazine*[4], o desenvolvimento de projetos de pesquisa direcionados para esta área é suficientemente desafiante para captar o interesse de diversos perfis de aluno na UFABC (sem discriminação de género), interessados em estudar ligações interessantes entre matemática, física e computação.

© *Nelson Faustino (orientador)*

[3] Link para IBM Q: `https://www.research.ibm.com/ibm-q/technology/experience/`

[4] vide `https://www.quantamagazine.org/tag/the-future-of-quantum-computing/`

2 Introdução

2.1 Estado da Arte

A computação quântica é a ciência que estuda as aplicações das teorias e propriedades da mecânica quântica na Ciência da Computação. Na computação clássica o computador é baseado na arquitetura que faz uma divisão entre elementos de processamento e armazenamento de dados, que possui processador e memória destacados por um barramento de comunicação, sendo seu processamento sequencial.

Entretanto os computadores atuais possuem limitações, como por exemplo na área de *Inteligência Artificial* (IA) onde não existem computadores com potência ou velocidade de processamento suficiente para suportar uma IA avançada. Dessa forma surgiu a necessidade da criação de um computador alternativo dos usuais que resolvesse problemas de IA, ou outros como a fatoração de números primos muito grandes, logaritmos discretos e simulação de problemas da Física Quântica.

Em 1965, Gordon (cf. [17]) faria uma predição de que a quantidade de transistores em um circuito integrado cresceria exponencialmente de modo que, a cada dois anos, a quantidade de transistores em um circuito integrado dobraria. Tal predição acabou tornando-se a lei de Moore no qual tornou-se comum dizer que a cada dois anos a velocidade de um computador dobraria. Mais tarde, esta viera a ser corrigida por David House (cf. [19]) ao dizer que a velocidade dobra a cada 18 meses.

Entretanto, por limitações física, é aguardado que a lei de Moore deixe de valer, fato este que já vem acontecendo com o aumento na dificuldade de reduzir o tamanho dos transistores (cf. [27]). Deste modo, incita-se o surgimento de novos métodos para que a computação continue tendo um crescimento e expandindo sua velocidade.

Por outro lado, em 1982, Richard Feynman (cf. [7]) sugeria em sua memorável expressão: "a natureza não é clássica, e se você quer fazer uma simulação da natureza, você deveria fazer isso com a mecânica quântica". Neste, Feynman propõe que, para que sejamos capazes de simular a natureza em sua verdadeira forma, deveríamos ter algo capaz de efetuar cálculos e que funcione da mesma maneira em que a natureza.

Em 1999, Peter Shor (cf. [26]), precedido por outros cientistas, torna a ideia do computador quântico ser um sucessor ao computador clássico, já em crescente difusão, ainda mais importante. Fazendo uso de um modelo teórico de computador que segue a mecânica quântica, este desenvolve um algoritmo para fatoração em números primos. Além de rodar em tempo polinomial, este se apresentou uma potencial solução ao sistema RSA (cf. [23])de criptografia.

A IBM, entre outras empresas, vem trabalhando em criar um computador quântico que atinja a chamada vantagem quântica (estado no qual

o computador quântico será superior ao clássico em diversas aplicações). Recentemente, em um encontro da American Physical Society, a IBM anunciou, pela terceira vez consecutiva, ter dobrado o "volume quântico"(meio de medir performance) de seu computador quântico System Q One (cf. [34]), seguindo assim, de certo modo, a lei de Moore.

2.2 Organização do Relatório

Os temas tratados ao longo deste relatório foram organizados do seguinte modo:

- No capítulo 3, é realizada a apresentação da mecânica quântica, tratando de temas como a equação de onda, sua interpretação probabilística (Bohr) e sua solução, o príncipio da incerteza de Heisenberg e o formalismo de Dirac, por sua vez, será utilizado de maneira abrangente neste trabalho. Também são definidos Operadores Hermitianos, quais serão utilizados, posteriormente, ao tratar de operadores quânticos.

- No capítulo 4, inicia-se propriamente a apresentação de tópicos da computação quântica. De começo, é apresentado o qubit (análogo quântico do bit) e deduzimos a equação de seu estado a partir da esfera de Bloch. São apresentados também portas quânticas, quais fundamentam as operações e transformações aos estados dos qubits, assim como sistemas multi-qubits.

- No capítulo 5, são apresentados dois algoritmos populares ao campo da computação quântica. Em cada um dos dois, é apresentado, de modo detalhado, o raciocínio e a maneira com que as propriedades quânticas, tais como interferência e sobreposição de estados, são utilizadas a favor do algoritmo.

- Por fim, no capítulo 6, é tratado sobre o algoritmo de Shor. Juntamente a este, são também explicadas as séries e as transformadas de Fourier, incluindo a Transformada Quântica de Fourier, qual será utilizada pelo próprio Shor em seu algoritmo.

3 Introdução à mecânica quântica

Este capítulo é fundamentado pelo livro de Griffiths (cf. [9]), complementado com as referências bibliográficas [6, 21, 22].

3.1 Equação de onda

De modo análogo à mecânica clássica, podemos estudar propriedades dinâmicas de corpos. No caso da mecânica quântica, nosso objeto de estudo são partículas microscópicas. Para realizarmos os estudos destas propriedades, iremos inicialmente determinar a função que nos possibilita um acesso a diversas grandezas. Esta é chamada de função de onda.

A função de onda, cujo símbolo é $\Psi := \Psi(x,t)$, corresponde à solução da equação de Schrödinger, qual é escrita por

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} + V\Psi, \tag{1}$$

sendo $i = \sqrt{-1}$, $\hbar$ a constante de Planck e $V := V(x)$ a função de potencial.

3.1.1 Interpretação da função de onda

A função de onda nos fornece curvas, o que, a principio, não aparenta nos dar nenhuma informação que possamos interpretar com clareza quanto a grandeza em questão, diferente da mecânica clássica onde as soluções são bem definidas a um ponto. Entretanto, através da interpretação estatística de Born (cf. [9, p. 2 do Capítulo 1]), a função de onda (Ψ) seria como uma função distribuição de probabilidade enquanto $|\Psi(x,t)|^2$ é a densidade de probabilidade cumulativa contínua (na variável x).

Tal modo de interpretar a função de onda nos leva a um certo indeterminismo, visto que a partícula pode, a partir da interpretação de Born, assumir qualquer valor x, dado sua devida probabilidade.

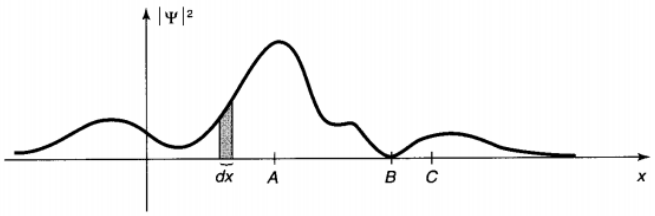

Figura 1: Aqui vemos uma função de onda onde o valor de x (grandeza) mais provável encontra-se em $x = A$. Fonte: GRIFFITHS, c2011, p. 3

Apesar de termos essa indeterminação a partir dos cálculos matemáticos, podemos realizar uma medição em uma grandeza, como, por exemplo, sua posição, nos permitindo assim descobrir onde a partícula realmente está de

maneira empírica. Consequentemente, já que sabemos onde nossa partícula está, isto é, temos 100% de certeza sobre sua localização, então nossa função de onda representada na figura 1 já não condiz mais com a realidade, visto que neste, a posição da partícula não está claramente definida. Deste modo, introduz-se o colapso da função de onda, no qual ela deixa, após a medição, de ser uma função esparsa e torna-se apenas um único pico no valor mensurado C.

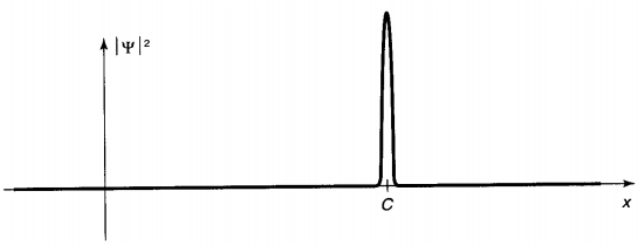

Figura 2: colapso da função de onda $|\Psi(x,t)|^2$, causado pela medição. Fonte: GRIFFITHS, c2011, p. 5

Novas medições realizadas no mesmo sistema irão resultar sempre no mesmo valor C, visto que, agora, este é o único valor no qual $|\Psi(x,t)|^2 > 0$.

3.2 Formalismo

3.2.1 Notação de Dirac

A função de onda, representando um estado, é um vetor, cuja representação é dada pela notação de Dirac. Neste, os vetores são chamados de ket (vetor coluna) $|\Psi\rangle$, onde Ψ representa um estado, conforme dito sobre Ψ sendo a função de onda. Para todos os vetores ket, existe o bra $\langle\Psi|$ que é o complexo conjugado transposto (operador adjunto ou conjugado Hermitiano) de Ψ, isto é, Ψ^* do estado a ser representado.

A interação entre os dois é chamada de bra-ket, e é escrita como $\langle\Psi|\Psi\rangle$ e este é o produto interno no espaço de Hilbert[1], ou seja,

$$\langle\Psi|\Psi\rangle = \int |\Psi(x,t)|^2 dx, \tag{2}$$

onde $|\Psi(x,t)|^2 := \Psi^*(x,t)\Psi(x,t)$ corresponde à função densidade de probabilidade (na variável x).

No caso de termos um operador $\hat{O}$ agindo em um ket, este nos fornece outro ket, ou seja

$$\hat{O}\,|\Psi\rangle = |\Phi\rangle$$

[1]O espaço de Hilbert é um espaço vetorial abstrato que generaliza a noção do espaço Euclidiano. Com dimensão infinita, este é dito completo e possui produto interno. As funções de onda estão contidas no espaço de Hilbert.

E o valor esperado de um operador, é escrito como

$$\begin{aligned} \langle \Psi | \hat{O} | \Psi \rangle &= \int \Psi^*(x,t) \hat{O} \Psi(x,t) dx \\ &= \int \Psi^*(x,t) \Phi(x,t) dx. \end{aligned}$$

3.2.2 Operadores

Nossas funções de onda são vetores no espaço de Hilbert e representam estados de nossa partícula, que analogo à mecânica clássica seriam como a posição e o momento, por exemplo.

Vamos agora ver as grandezas físicas que são representados por operadores. Na mecânica clássica, possuímos funções e relações entre grandezas que usam os estados como argumento e tem como resultado um valor, tal como posição, momento, energia cinética e outros.

Porém, diferente da mecânica clássica, nossa grandeza somente terá um valor condizente com a teoria se, e somente se estivermos lidando com um autoestado. Deste modo, teremos um autovalor sendo atribuído à grandeza em questão. Vamos portanto retomar algumas ideias sobre autoestados.

Dos conceitos de algebra linear, uma transformação linear (matriz) possui alguns vetores, chamado autovetor, no qual a transformação opera ao entorno, isto é, estes não são rotacionados.

Estes vetores são importantes para a álgebra linear pois nos permite entender melhor a transformação em questão. Contudo, na mecânica quântica, estes tem uma função ainda mais importante.

Foi dito até agora que, a partir de uma função de onda, podemos descobrir informações quanto a uma grandeza em específico. Entretanto, para tal, devemos aplicar o que chama-se operador. Por exemplo, se desejarmos saber quanto momento de uma partícula, deve-se relacionar com o estado da partícula (Ψ) a uma transformação linear (mais comumente chamada de operador) referente à grandeza momento.

Estes operadores são lineares, e, com isso, carregam autoestados. Vimos na seção 3.1.1 que, segundo a interpretação de Born, $|\Psi(x,t)|^2$ nos fornece uma função no qual valores maior do que 0 são possíveis resultados que obteremos ao realizar uma medição. Portanto, o que a densidade de probabilidade $|\Psi(x,t)|^2$ nos fornece é justamente a probabilidade de obtermos determinado autoestado relacionado ao operador aplicado (transformação linear para obtermos uma função da grandeza) em questão. Para ser mais claro, nossa função de onda Ψ é transformada, a partir de um operador, a uma combinação linear das autofunções do operador em questão. Deste modo, quando temos mais de uma autofunção atrelada à função de onda, após a transformação linear, cuja probabilidade é maior do que 0, dizemos, devido a interpretação de Born e comprovado através do experimento da

dupla fenda, que nossa partícula está em uma sobreposição de estados, propriedade esta que será melhor tratada em breve.

Com isso, podemos resumir que, de maneira geral, um operador é uma transformação linear que opera na função de onda, alterando-a de tal modo que ela seja um função de estado de uma grandeza física. A notação utilizada é que um operador $\hat{O}$ representa a grandeza O, podendo ser esta exemplificada como o momento, a velocidade ou qualquer outra grandeza conhecida nos fenômenos clássicos. Podemos ver a atuação de um operador $\hat{O}$ em um estado $\Psi(x,t)$ ao fazermos, conforme a notação de Dirac, $\hat{O}\left|\Psi\right\rangle = \left|\Phi\right\rangle$.

A maior parte dos operadores utilizados na mecânica quântica são de um tipo especial chamados Hermitiano. Um operador $\hat{H}$ é considerado Hermitiano quando ele pode ser aplicado para qualquer um dos vetores, sem alterar a igualdade, isto é,

$$\left\langle f\middle|\hat{H}g\right\rangle = \left\langle \hat{H}f\middle|g\right\rangle \tag{3}$$

Esta descrição define operadores Hermitianos, porém, estes possuem algumas propriedades importantes de comentar:

1. Seus autovalores são sempre reais, apesar de sua autofunção poder ser complexa.

2. Suas autofunções podem sempre serem escolhidas de modo a ser normalizadas e ortogonais. Deste modo, as autofunções formam um conjunto completo, ou seja, qualquer função pode ser escrita como combinação linear dos autovetores.

3. No caso de produto linear, a igualdade $\left\langle f\middle|\hat{H}g\right\rangle = \left\langle g\middle|\hat{H}f\right\rangle^{*}$, também é verdade e gera um número real.

4. Autovalores distintos associados a autovetores distintos são ortogonais entre si.

Podemos provar a propriedade 4, exceto para o caso em que nossos estados são degenerados, isto é, duas autofunções distintas possuem o mesmo autovalor.

Vamos assumir que $\hat{H}$ seja um operador Hermitiano com duas autofunções, tal que,

$$\hat{H}\Psi_1 = \lambda_1\Psi_1 \quad \text{e} \quad \hat{H}\Psi_2 = \lambda_2\Psi_2.$$

Podemos então, de acordo com a definição dos operadores Hermitianos, escrever $\left\langle\Psi_2\right|\hat{H}\left|\Psi_1\right\rangle$ de duas formas distintas:

1. $\left\langle\Psi_2\middle|\hat{H}\Psi_1\right\rangle = \lambda_1\left\langle\Psi_1|\Psi_2\right\rangle$

2. $\left\langle \Psi_2 \middle| \hat{H}\Psi_1 \right\rangle = \langle H\Psi_2 | \Psi_1 \rangle = \lambda_2 \langle \Psi_1 | \Psi_2 \rangle$

Igualando as duas últimas equações e colocando-as no mesmo lado da igualdade, temos

$$0 = (\lambda_2 - \lambda_1) \langle \Psi_1 | \Psi_2 \rangle$$

Visto que $\lambda_1 \neq \lambda_2$, por hipótese, a única forma dessa equação ser verdade é se o produto interno entre Ψ_1 e Ψ_2 for igual a zero, o que implica que Ψ_1 é ortogonal a Ψ_2, tal como pretendido.

4 Circuitos Quânticos

Agora que já temos uma boa visualização sobre o que é a mecânica quântica e sua formulação matemática, passemos de seguida à formulação de alguns conceitos basilares de computação quântica. Para demais detalhes, vide também [4, Capítulo 2].

4.1 Quantum bit

No computador clássico, a unidade elementar de informação é chamada de bit, qual pode assumir dois valores, 0 ou 1. Já no computador quântico, a unidade elementar de informação é chamada de qubit (quantum bit). Em termos da notação de Dirac introduzida em [6], um qubit pode ser escrito como $\alpha\,|0\rangle + \beta\,|1\rangle$ (cf. [16])(p. 38-40), onde α e β são escalares complexos, e $|0\rangle$ e $|1\rangle$ são autoestados do operador (lê-se soluções da equação de Schrödinger) (cf. [9])(p. 105) **spin**[5]. Portanto, um qubit é escrito como um vetor coluna $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ cuja norma é igual a 1.

Um dos benefícios que o qubit, por usar a grandeza do spin, nos fornece é que este pode existir em um estado como uma combinação linear de seus autovetores (spin up e spin down), fazendo uso da mesma propriedade apresentada referente a função de onda estar definida para vários valores de seu domínio:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Esta grande diferença entre bit e o qubit é uma das principais razões que faz o computador quântico ser potencialmente mais rápido do que o clássico, motivo qual será explicado posteriormente.

Tal propriedade de sistemas quânticos é chamada de sobreposição. A sobreposição, como já descrito brevemente ao fim da seção de operadores, é consequente ao fato da função de onda ser resultado de uma combinação linear das autofunções referente a grandeza a ser representada, no caso spin. Portanto, sendo o spin definido em dois estados (spin up e spin down), estes são seus autovetores. Com isso, podemos atribuir 0 e 1 para estes e utiliza-los como valores lógicos de um qubit.

Com isso, podemos descrever o estado de um qubit em sobreposição como

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \quad \alpha, \beta \in \mathbb{C} \qquad \text{e} \quad |\alpha|^2 + |\beta|^2 = 1.$$

[5]Spin é uma propriedade quântica inerente às partículas, qual não tem análogo clássico. Neste, uma partícula pode assumir dois estados, spin up ou spin down (cf. [16])(p. 40). Por apresentar duas possibilidades, este pode ser usado para representar estados lógicos como 0 e 1

Apesar de possuirmos nossa informação de maneira dividida/ambígua, como promovido pela sobreposição, a mesma não pode ser acessado por inteiro devido ao colapso da função de onda. Ou seja, por mais que utilizemos a sobreposição durante a computação, quando efetuamos uma medição, o estado do qubit vai colapsar em $|0\rangle$ com probabilidade $|\alpha|^2$ ou $|1\rangle$ com probabilidade $|\beta|^2$, alterando portanto o estado do qubit para um de seus autovetores.

Para podermos representar pictoricamente a noção qubit, podemos recorrer à parametrização das quantidades $|\alpha|$ e $|\beta|$ em termos de coordenadas polares tal como aparece ilustrado na figura 3, em que $|0\rangle$ e $|1\rangle$ são identificados como vetores da base canónica de $\mathbb{R}^2$.

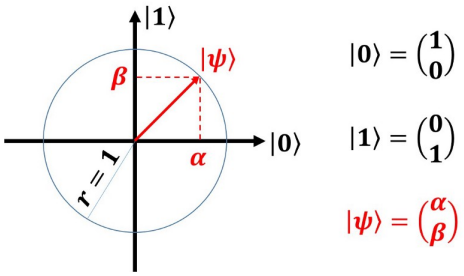

Figura 3: Qubit representado para quando α e β são números reais, i.e., representado sem levar em conta o eixo dos imaginários.[7]

Para o caso particular de escolhermos

$$\alpha = \cos\left(\pm\tfrac{\pi}{4}\right) = \tfrac{1}{\sqrt{2}} \text{ e } \beta = \sin\left(\pm\tfrac{\pi}{4}\right) = \pm\tfrac{1}{\sqrt{2}}$$

a combinação linear acima corresponde a

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{ou} \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}},$$

onde $\sqrt{2}$ corresponde ao fator de normalização associado aos ângulos de rotação $\theta = \pm\frac{\pi}{4}$ no plano cartesiano.

Para que tenhamos um completo entendimento sobre os qubits, teremos de recorrer à parametrização de $\alpha, \beta \in \mathbb{C}$ em termos de coordenadas esféricas. Este modelo de representação é conhecido na literatura por esfera de Bloch (cf. [8]):

[7] Disponível em https://medium.freecodecamp.org/almost-everything-you-ever-wanted-to-know-about-quantum-computers-5ee6bc2f40ba. Acesso fev. 2019

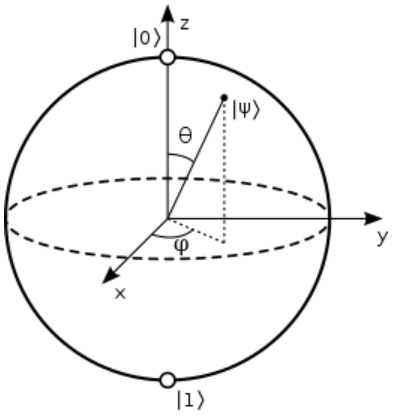

Figura 4: Representação do qubit na esfera de Bloch.[8]

A demonstração desta correspondência assenta essencialmente na representação polar de um número complexo $a + ib$ na forma

$$a + bi = re^{i\theta},$$

onde $r^2 = a^2 + b^2$ e $\cos(\theta) = \dfrac{a}{r}$.

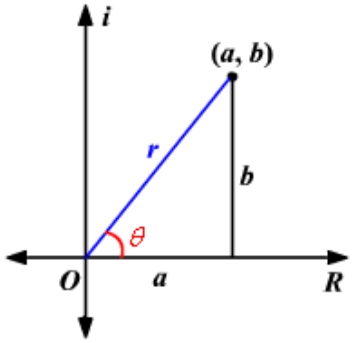

Figura 5: Representação polar do número complexo $a + bi \in \mathbb{C}$. [9]

Em concreto, se considerarmos as parametrizações

$$\alpha = |\alpha|e^{i\phi_0} \text{ e } \beta = |\beta|e^{i\phi_1},$$

obtemos a sequência de igualdades:

$$\begin{aligned} |\psi\rangle &= |\alpha|e^{i\phi_0}|0\rangle + |\beta|e^{i\phi_1}|1\rangle \\ &= e^{i\phi_0}\left(|\alpha|\,|0\rangle + |\beta|e^{i(\phi_1-\phi_0)}\,|1\rangle\right). \end{aligned}$$

[8]Disponível em https://medium.com/@jackceroni/quantum-computing-a-brief-introduction-74b5084af6f0. Acesso fev. 2019.

[9]Disponível em https://precalculusstudy.weebly.com/lesson-65.html. Acesso fev. 2019.

Adicionalmente, da condição

$$|\alpha|^2 + |\beta|^2 = 1,$$

retiramos que

$$|\alpha| = \cos(\theta) \quad \& \quad |\beta| = \sin(\theta).$$

Sendo $e^{i\phi_0}$ um fator de fase que satisfaz a condição $|e^{i\phi_0}| = 1$, podemos descartá-la, pois não é observável. Neste sentido, podemos considerar a seguinte representação reduzida (ou 'normalizada')

$$|\psi\rangle = \cos(\theta)\,|0\rangle + \sin(\theta)e^{i\phi}\,|1\rangle\,, \quad \text{onde} \quad \phi = \phi_1 - \phi_0.$$

Finalmente, fazendo as substituições $x + iy = \sin(\theta)e^{i\phi}$ e $z = \cos(\theta)$, tem-se que parametrização

$$\begin{cases} x = \sin(\theta)\cos(\phi) \\ y = \sin(\theta)\sin(\phi) \qquad (0 \leq \phi \leq 2\pi, \quad 0 \leq \theta \leq \pi) \\ z = \cos(\theta) \end{cases}$$

nos dá correspondência entre a reprsesentação acima e a esfera de Bloch.

Provámos assim que o qualquer qubit pode assumir qualquer estado dentro da esfera unitária, como pretendido.

4.2 Portas quânticas de 1 qubit

Já vimos que um qubit pode ser representado por $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, podendo este ser escrito com um vetor coluna do tipo

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} := \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \alpha, \beta \in \mathbb{C}$$

em que este pode assumir qualquer vetor dentro da esfera de Bloch.

Vamos nesta seção, a partir do estudo das refs. [28, 16, 12]) ver como é feita a manipulação dos qubits, isto é, iniciar os estudo aos elementos lógicos que compõem um computador quântico. Demais detalhes podem ser também encontrados em [4, Seção 2.3].

As portas lógicas da computação quântica, são bem diferentes daquelas utilizadas no computador clássico, isso se deve a dois pontos:

- Portas quânticas são reversíveis;
- Tem a forma de *matrizes unitárias*[10].

[10] Matrizes unitárias A são matrizes que satisfazem a igualdade $A^*A = I$, onde I denota a matriz identidade.

Portanto, quando um qubit passa por uma porta quântica, este tem seu módulo conservado, e caso seja novamente aplicado a transformação com a mesma porta, o estado do sistema retornará ao seu estado inicial, isto é, Sendo A uma porta quântica, ao fazermos a sequência de tranformações

$$\vec{v} \longrightarrow A\vec{v} \longrightarrow A^*A\vec{v}$$

estamos na verdade fazendo $I\vec{v}$, ou seja, multiplicando $\vec{v}$ por uma matriz identidade I.

Transformações unitárias podem serem vistas como rotações em torno da esfera de Bloch, deste modo, as portas quânticas operam logicamente por manipular a amplitude dos autovetores de um sistema quântico.

Vamos abaixo ver algumas portas quânticas que operam em apenas um único qubit:

- **Pauli X Gate**

  A porta Pauli X efetua, em análogo às portas lógicas clássicas, uma negação no estado do qubit, isto é, $X|0\rangle = |1\rangle$, ou seja, rotação em torno do eixo X.

  No computador clássico, temos a operação lógica **NOT** que é análoga a porta Pauli X, por isso, esta muitas vezes é chamada de quantum NOT gate.

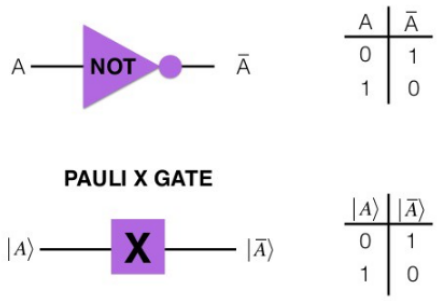

Figura 6: Comparação entre Pauli X e porta NOT.[11]

  Sua forma matricial

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

- **Pauli Y Gate**

  A porta Pauli Y efetua uma rotação em torno do eixo Y transformando $|0\rangle$ em $i\,|1\rangle$ e $|1\rangle$ em $-i\,|0\rangle$, ou seja, é uma porta NOT (ou Pauli X), porém contendo i como escalar.

[11] Disponível em https://towardsdatascience.com/demystifying-quantum-gates-one-qubit-at-a-time-54404ed80640. Acesso fev. 2019.

Sua forma matricial é escrita como

$$X = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

- **Pauli Z Gate**

  Como todos as outras portas de Pauli, a Z gate opera uma rotação em torno do eixo Z. Perceba, porém, que ao efetuar uma rotação em torno do eixo Z, não estamos a alterar a probabilidade de encontrar um dos estados, entretanto, este altera a relação $|0\rangle + |1\rangle$ para $|0\rangle - |1\rangle$. Deste modo somos introduzidos à propriedade chamada de fase, que será melhor explicado ao tratarmos sobre algoritmo quântico.

  Sua forma matricial é

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

De forma geral, podemos visualizar cada uma das transformações de Pauli como na imagem a seguir:

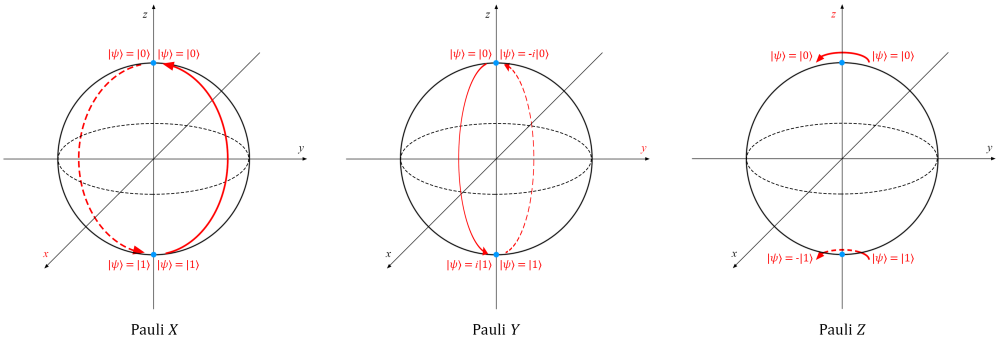

Figura 7: As transformações Pauli-X, Pauli-Y e Pauli-Z, respectivamente.[13]

- **Porta de Hadamard**

  A porta de Hadamard é responsável por transformar um estado puro em uma sobreposição de estados totalmente equilibrada, isto é, ambos estados com iguais amplitudes.

  Podemos, por exemplo, aplicar a transformação de Hadamard em um qubit no estado puro $|0\rangle$ fazendo $|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Perceba que ambos os estados possuem a mesma amplitude $\frac{1}{\sqrt{2}}$, deste modo, ao ser medido,

[13] Disponível em https://blogs.msdn.microsoft.com/uk_faculty_connection/2018/02/26/quantum-gates-and-circuits-the-crash-course/. Acesso fev. 2019.

o qubit tem 50% de chance de ser encontrado em $|0\rangle$ e 50% de chance de ser encontrado em $|1\rangle$.

Podemos escrever esta transformação de duas formas:

– Com a notação de Dirac

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

Que utiliza o produto interno com o qubit a ser operado.

– Com a forma matricial

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

Podemos visualizar um exemplo onde aplicamos H a $|0\rangle$,

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Deste modo, temos

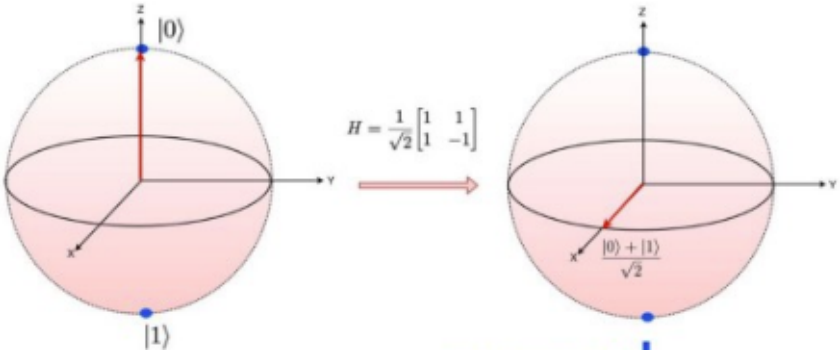

Figura 8: Exemplo da transformação de Hadamard onde o estado é representado pela seta vermelha.[15]

Este acaba sendo de extrema importância para os algoritmos quânticos, pois, além de ser capaz de criar uma sobreposição (propriedade que permite simultaneidade), este, diferente do processo de medição, é capaz de perceber diferentes fases (ao aplicar hadamard a um qubit em sobreposição), fases estas que serão de grande utilidade, como veremos nos estudos de algoritmos quânticos.

- **Porta de Fase** Assim como vimos na porta Pauli Z, a porta de fase realiza rotações ao entorno do eixo Z.

[15]Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-8996b667d256. Acesso fev. 2019.

Com esta, diferente da Pauli Z, podemos escolher quantos radianos queremos que nosso vetor rotacione.

Na notação de Dirac (cf. [6]), pode ser escrita como:

$$R_\theta = |1\rangle\langle 0| + e^{i\theta}|0\rangle\langle 1| \tag{4}$$

Ou, na forma matricial:

$$R_\theta = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$$

4.3 Produto Tensorial

Até o momento, falamos apenas sobre como pensar e lidar com um único qubit, portanto, veremos agora sistemas de varias partículas.

Podemos, por intuição, pensar que dado duas partículas com respectivos estados $\vec{v} \in V$, e $\vec{w} \in W$ o sistema formado por elas seja $(\vec{v}, \vec{w})$. Porém, apesar deste nos dizer o estado das duas partículas, o mesmo não representa o estado geral do sistema. Deste modo, introduz-se o **Produto Tensorial**.

Para que possamos representar o estado do sistema composto por $\vec{v}$ e $\vec{w}$, utiliza-se o produto tensorial, escrito como $\vec{v} \otimes \vec{w}$. Este elemento será visto como um vetor em um novo espaço $V \otimes W$ (cf. [16])(p. 33 - 35).

Suponha que

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \end{bmatrix} |\phi\rangle = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

então $|\psi\rangle \otimes |\phi\rangle$ é um vetor construído como

$$|\psi\rangle \otimes |\phi\rangle = \begin{bmatrix} \psi_1\phi_1 \\ \psi_1\phi_2 \\ \psi_2\phi_1 \\ \psi_2\phi_2 \end{bmatrix}$$

Tal formalismo traz consigo as seguintes características:

- Escalar um dos vetores do produto tensorial é equivalente a escalar todo o produto tensorial, isto é,

  $(av) \otimes w = v \otimes (aw) = a(v \otimes w), \quad \forall\ a \in \mathbb{C}$

- Se temos vetores em sobreposição, então $\vec{v} \otimes \vec{w}$ também estará

  $(v_1 + v_2) \otimes w = v_1 \otimes w + v_2 \otimes w$

  $(v_1 + v_2) \otimes (w_1 + w_2) = v_1 \otimes w_1 + v_1 \otimes w_2 + v_2 \otimes w_1 + v_2 \otimes w_2$

Portanto, o espaço $V \otimes W$ é o espaço formado pela combinação linear dos elementos na forma $u \otimes v$.

Mais precisamente, sendo $v_1, v_2 \in V$, e $w_1, w_2 \in W$, geraremos os vetores $v_1 \otimes w_1$, $v_2 \otimes w_2 \in V \otimes W$. Caso ainda v_1, v_2 e w_1, w_2 formem uma base de seu espaço, então $v_1 \otimes w_1$, $v_2 \otimes w_2$ formara uma base para $V \otimes W$

Quanto aos operadores, estes continuam a operar apenas nos vetores que estavam no mesmo espaço antes do produto tensorial, isto é, seja T um operador em V, e S um operador em W, temos que

$\mathrm{T}\ (\otimes w) = (\mathrm{T}v) \otimes w$,

e o produto tensorial entre as duas transformações T $\otimes$ S, aplicado ao produto tensorial $v \otimes w$, é realizada como

$\mathrm{T} \otimes \mathrm{S}\ (v \otimes w) = \mathrm{T}v \otimes \mathrm{S}w$.

Na notação de Dirac, representamos um produto tensorial de um sistema de n qubit como $|vw...u\rangle$, ou podendo ser escrito como $|v\rangle\,|w\rangle\,...\,|u\rangle$ (cf. [25])(p. 66) carregando todas as propriedades acima descritas.

4.4 Portas quânticas para n qubits

Visto que agora já fomos introduzidos à sistemas de vários qubits, veremos como trabalhar com sistemas multi-qubits, se baseando nos trabalhos (cf. [28])(p. 17-18) e (cf. [13]).

Antes de entrarmos nas portas utilizadas para n qubit, vamos ver como é representado um circuito quântico.

Um circuito quântico é composto por **registradores**, portas quânticas e um operador de medição.

Podemos ver bem essa construção a partir da figura abaixo:

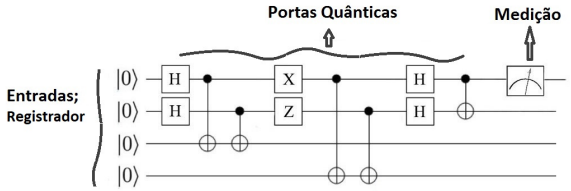

Figura 9: Exemplificando um circuito quântico, apontando seus componentes.[17]

Registrador é o conjunto de nossos qubits. Este é representado pelas linhas horizontais vista em nosso circuito quânticos. Portanto, dado um conjunto de n qubits $|0\rangle\,|0\rangle\,...\,|0\rangle$, nosso registrador representará o estado

[17]Modificado pelo autor, com base na imagem disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-8996b667d256. Acesso fev. 2019.

$|00...0\rangle$, isto é, nosso registrador é feito pelo produto tensorial de cada qubit, fazendo deste um vetor. Note que nossos qubits podem estar em uma sobreposição, deste modo, o registrador poderá ser visto como vetor composto pela amplitude de cada um de seus autovetores, que é análogo a dizer que este é feito pela combinação linear de seus autovetores.

Vamos agora ver as portas quânticas utilizadas para sistemas de vários qubits.

- **Porta C-NOT**

  A porta C-NOT, ou Controlled NOT, opera com dois qubits. Um é utilizado como controle, enquanto o outro é quem recebe a operação NOT (caso o qubit de controle seja $|1\rangle$), chamado target.

  Podemos visualizar seu funcionamento como a seguir

Figura 10: Exemplo de aplicação da porta C-NOT para o caso onde o controle é 0 e 1.[19]

  De maneira mais clara, a porta C-NOT inverte o segundo qubit (target) quando o primeiro (control) é igual a 1. Deste modo, podemos escreve-lo de forma lógica da seguinte maneira

$$|x\rangle \otimes |y\rangle = |x\rangle \otimes |x \oplus y\rangle \tag{5}$$

  onde $\oplus$ representa o XOR (exclusive or).

  Sua forma matricial

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

  Podemos verificar seu funcionamento em uma sobreposição, utilizando um qubit auxiliar = $|1\rangle$ para o controle:

[19]Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso fev. 2019.

$$(1, y) \rightarrow (1, y \oplus 1)$$
$$|1\rangle(|0\rangle - |1\rangle) \rightarrow |1\rangle(|1\rangle - |0\rangle)$$

Figura 11: Aplicando CNOT em um sistema $|1\rangle(|0\rangle - |1\rangle)$.[20]

- **Porta Controlled U** Esta é uma generalização da CNOT, vista anteriormente.

  Podemos perceber que a CNOT realiza o mesmo que a Pauli X, porém utilizando uma porta que controle a aplicação.

  Se tivermos um olhar atento, veremos que se formarmos um quadrado em torno dos últimos 4 elementos, teremos uma porta Pauli X.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Figura 12: Representação matricial da porta CNOT, com foco nos 4 últimos elementos. Fonte: Elaborado pelo autor.

  Com isso, surge a Controlled U, onde mantém-se todos os elementos iguais à CNOT, porém, altera-se os 4 últimos, como em destaque na figura acima.

  Portanto, podemos descreve-la como um operador onde, tendo determinado valor para o qubit de controle, aplica-se ao target a operação U, qualquer que seja U. Assim sendo, esta é representada como

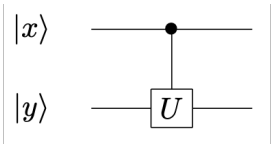

Figura 13: Circuito contendo uma Controlled U.[21]

- **Porta de Toffoli**

  A porta de Toffoli é uma porta controlada onde os dois primeiros qubits servem como controle e o terceiro é o target, ou seja, e uma CNOT com um Controlled a mais, ficando assim CCNOT. Ao termos

[20] Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso fev. 2019

[21] Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso fev. 2019

$|1\rangle$ para os dois primeiros, o terceiro qubit é invertido. Ou seja, este simula uma porta NAND.

Como a NAND pode expressar todas as funções booleanas, logo a porta de Tofolli também o faz. Portanto, podemos com esta realizar todas as operações que as portas lógicas clássicas realizam.

Pode ser visualizada como: Sua forma matricial é

$$(a, b, c) \rightarrow (a, b, c \oplus ab)$$

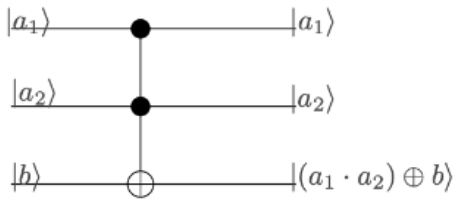

Figura 14: Exemplificando o circuito contendo uma porta de Toffoli.[22]

$$CCNOT = \begin{Bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{Bmatrix}$$

- **Swap gate**

  A porta de permutação, opera em trocar o estado de dois qubits entre si, ou seja, sendo $|0\rangle$ e $|1\rangle$, ao aplicar a Swap gate, ficamos com $|1\rangle$ e $|0\rangle$.

  É escrita como a matriz

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

[22]Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso fev. 2019

- **Hadamard e multipla sobreposição**

  Vimos na seção 4.2 que podemos colocar um qubit em uma sobreposição de estados ao aplicar a porta de Hadamard.

  Apesar da porta de Hadamard operar em um único qubit, podemos aplicar n Hadamard em n qubits, formando n estados em sobreposição.

  Para que fique mais claro, vamos ilustrar um sistema de 3 qubits, todos no estado inicial $|0\rangle$. Aplicamos uma Hadamard a cada um deles, obtendo a sobreposição:

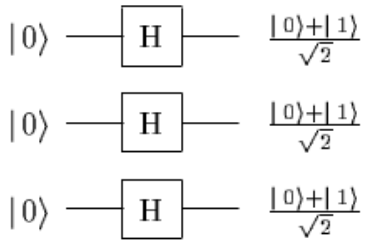

Figura 15: A esquerda, o qubit no estado inicial zero, sofrendo a transformação de Hadamard e resultando em uma sobreposição.[24]

Agora, temos um sistema $\frac{1}{\sqrt{2^3}}((|0\rangle > + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle))$, e como ja vimos anteriormente, o produto tensorial possui uma propriedade distributiva.

Teremos então uma sobreposição dos seguintes estados:

$$\psi = \frac{1}{\sqrt{2^3}}(|000\rangle + |001\rangle + |010\rangle + |001\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle) \quad (6)$$

Portanto, por simplicidade, podemos escrever uma operação de Hadamard em vários qubits. Adotando $|x\rangle$ como o sistema que contém todos os nossos vetores ($|x\rangle = |000...0)\rangle$, temos

$\mathrm{H} \otimes \mathrm{H} \otimes ... \otimes \mathrm{H}|x\rangle = \psi$, para o ψ demonstrado acima.

portanto, podemos reescrever toda a operação como

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in 0,1^n} (-1)^{xy} |y\rangle \quad (7)$$

[24]Disponível em https://www.quantiki.org/wiki/basic-concepts-quantum-computation. Acesso fev. 2019.

4.5 Emaranhamento

Outra propriedade proveniente de fenomenos quânticos é o emaranhamento quântico, também conhecido por entrelaçamento quântico. Uma das mais famosas propriedade quântica e considerada a propriedades menos intuitivas que esta apresenta, tem seu histórico quanto à negação de Einstein dado a proposição de tal fenômeno (cf. [31])(p. 24-25).

De maneira direta, o emaranhamento é a propriedade que partículas apresentam quando seus estados tornam-se interdependente, ou seja, o fato de uma partícula estar em uma posição x implica que outra partícula estará em outra posição y. O que mais incomodava Einstein quanto a esta propriedade é que, não importa quão longe uma partícula esteja da outra, contanto que elas estejam emaranhadas, qualquer mudança no estado de uma implicará na mudança do estado de outra. Deste modo, ocorre uma contradição quanto ao apresentado na relatividade especial, qual, em suma, apresenta que nada pode viajar mais rápido do que a velocidade da luz.

4.5.1 Verificando e construindo estados emaranhados

Dado $|a\rangle = \alpha|0\rangle + \lambda|1\rangle$ e $|b\rangle = \beta|0\rangle + \Lambda|1\rangle$ dizemos que ψ é um estado emaranhado se e somente se não existir $\psi = |a\rangle \otimes |b\rangle$ (cf. [35, p. 8]). Por exemplo, $|\psi\rangle = |0\rangle - |1\rangle$.

Este certamente não é um vetor resultante do produto tensorial entre $|a\rangle\, e\, |b\rangle$, portanto, este é dito ser um estado emaranhado. Por fim, vamos estudar a construção destes estados.

Podemos bem facilmente construir estados emaranhados ao aplicar apenas a porta CNOT, juntamente com a porta de Hadamard, criando um circuito na forma,

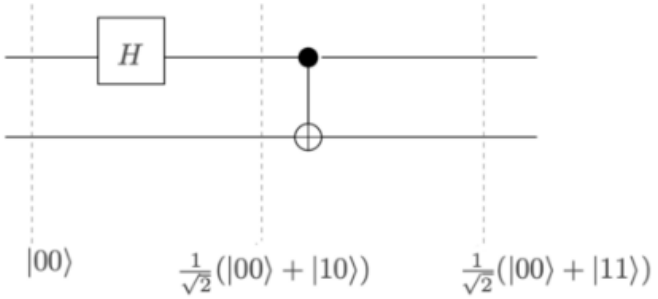

Figura 16: Circuito responsável por criar estados emaranhados.[25]

onde, como se pode ver, o sistema ao fim da operação não é equivalente a nenhum vetor de $H|0\rangle \otimes |0\rangle$, portanto, é um estado emaranhado.

[25]Disponível em https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso fev. 2019.

5 Algoritmos Quânticos

Agora que já temos uma boa base quanto ao formalismo e às propriedades necessária para construir um circuito quântico, vamos iniciar o estudo dos algoritmos quânticos.

5.1 Algoritmo de Deutsch-Jozsa

O algoritmo de Deutsch-Jozsa (cf. [5])foi um dos primeiros algoritmos a demonstrarem o potencial do computado quântico, apresentando maior velocidade em relação ao seu equivalente clássico.

Basicamente, o que este faz é, dado uma função $f : \{0,1\} \to \{0,1\}$, desejamos verificar se essa função é balanceada[26] ou constante, ou seja, para $x \in \{0,1\}$ (cf. [32, p. 1]), isto é

$$f(x) = 0, \forall x \in \{0,1\} \text{ ou } f(x) = 1 \ \forall \ x \in \{0,1\}.$$

Balanceada: $f(0) = 0, f(1) = 1$

Balanceada: $f(0) = 1, f(1) = 0$

Constante: $f(0) = 1, f(1) = 1,$

Constante: $f(0) = 0, f(1) = 0,$

Figura 17: A relação entre o dominio e a imagem de uma função que a determina como constante ou balanceada.[28]

Podemos ver que este é um problema bem simples, no qual podemos pensar em resolvê-lo classicamente ao testar a função para todas as entradas possíveis, isto é, sendo n o numero de bits (cada um podendo ser escrito como 0 ou 1), teriamos $2^{n-1} + 1$ "teste"da função. Computacionalmente falando, existe um certo gasto de tempo para efetuar tal processo.

Em computação quântica, por outro lado, podemos resolver o problema ao testar a função uma única vez, bastando utilizar o circuito a seguir:

[26] Toda a função balanceada é uma bijeção de $\{0,1\}$ em $\{0,1\}$.

[28] Disponível em https://medium.com/@jonathan_hui/qc-quantum-algorithm-with-an-example-cf22c0b1ec31. Acesso março 2019

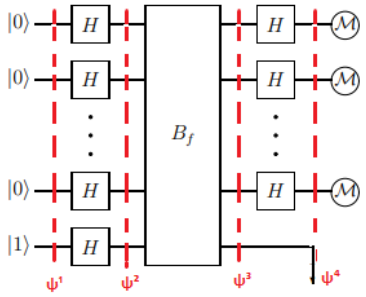

Figura 18: Circuito referente ao algoritmo de Deutsch-Jozsa. Fonte: University of Waterloo - Conteúdo expositivo para aulas.[30]

onde B_f é uma transformação definida como $B_f |x\rangle |b\rangle = |x\rangle |b \oplus f(x)\rangle$, para $|x\rangle = |000...0\rangle$ e $|b\rangle = |1\rangle$ (último qubit). Deste modo, estamos dizendo que esta transformação ocorre de maneira análoga a porta CNOT onde o estado do último qubit é depende do valor de $f(x)$.

Transformações como B_f são chamados de Oráculo, isto pois ela é composta por uma função $f(x)$ cujo comportamento não conhecemos. Ou seja, sempre que estivermos lidando com uma função de comportamento desconhecido, a chamaremos de oráculo.

Com base nos trabalhos de [32], [1] e [14] podemos resumir o algoritmo de Deutsch-Jozsa com os seguintes passos:

1. ψ_1 Inicia-se todos os qubits no estado $|0\rangle$, exceto pelo qubit auxiliar, qual é definido como $|1\rangle$. Deste modo temos $\psi_0 = |0000...1\rangle$.

2. ψ_2 Aplica-se uma porta de Hadamard para cada um deles, criando um sistema $\psi_1 = (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \otimes ... \otimes (|0\rangle - |1\rangle)$.

3. ψ_3 Agora, temos nosso registrador ψ_2, que, por sua vez, carrega todos os possíveis estados como apenas um único vetor, tendo sido operado por B_f. Neste ponto, ocorreu o que chamamos de processamento paralelo, onde com apenas uma única aplicação de B_f, conseguimos manipular todos os 2^n possíveis estados. Outro evento importantíssimo que ocorre nesta operação é o surgimento da interferência.

4. Por fim, aplicamos Hadamard ao registrador, resultando em um sistema composto por apenas um dos autovetores que antes estava em sobreposição.

5. Efetuamos a medição.

[30] Disponível em thttps://cs.uwaterloo.ca/ watrous/LectureNotes/CPSC519.Winter2006/05.pdf. Acesso março 2019.

Agora que já temos uma ideia do processo realizado no algoritmo de Deutsch-Jozsa (cf. [5]), vamos entender seu funcionamento de maneira mais aprofundada e detalhada, utilizando o que sabemos sobre a computação quântica e seu formalismo.

De maneira mais formal e detalhadamente explicada:

1. Iniciamos nosso registrador ψ^1 no estado $|0\rangle^{\otimes n} |1\rangle$

2. Aplicamos Hadamard a $|\psi_1\rangle$:

   $H^{\otimes n} |0\rangle^{\otimes n} \otimes H |1\rangle$, que podemos algebricamente escrever como

   $H^{\otimes n} |\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |0\rangle \, (\frac{|0\rangle - |1\rangle}{\sqrt{2}})$, onde n é o número de qubits.

   Com isso chegamos ao estado ψ_2

3. O oráculo B_f é aplicado em ψ_2 tal que

   $U_f |x\rangle |y\rangle = |x\rangle |y\rangle \oplus f(x) >$, onde x $= H^{\otimes n}|0>^{\otimes n}$, y $= H |1\rangle$ e $\oplus$ é equivalente a porta clássica XOR (exclusive OR).

   Podemos verificar o comportamento de U_f, ao verificar os possíveis resultados de $f(x)$.

   Vemos que para $f(x) = 0$, o segundo qubit permanece y, pois $0 \oplus (|0\rangle - |1\rangle)$ é equivalente a dizer $|0 \oplus 0\rangle - |1 \oplus 0\rangle$, onde seguindo a propriedade de $\oplus$ (para valores iguais, resultado $= 0$; para valores diferentes, resultado $= 1$), teremos $|0\rangle - |1\rangle$, ou seja,

   $$U_f : (x, y) \to |x\rangle \, \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{8}$$

   Para $f(x) = 1$, temos uma inversão no segundo qubit. Este fato pode ser comprovado utilizando a mesma linha de raciocínio vista para o caso anterior

   $U_f(x, y) \to |x\rangle \, \frac{(|1\rangle - |0\rangle)}{\sqrt{2}}$ que é equivalente a uma inversão na amplitude, onde, interpretando a esfera de Bloch, seria equivalente a um espelhamento com base no plano (x, y). Deste modo, podemos escrever U_f para $f(x) = 1$ como:

   $$- |x\rangle \, \frac{(|0\rangle - |1\rangle)}{\sqrt{2}} \tag{9}$$

   Ou seja, para $f(x) = 0$, temos o estado $|x\rangle \, \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$, e para $f(x) = 1$, temos $- |x\rangle \, \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$. Podemos, portanto, escrever uma equação equivalente a transformação observada:

   $(-1)^{f(x)} |x\rangle \, (|0\rangle - |1\rangle)$

Ou seja, U_f é equivalente a $(-1)^{f(x)}$

Podemos agora aplicar U_f em $H^{\otimes n}|\psi_1\rangle$, gerando $|\psi_2\rangle$, formando a equação:

$$U_f|\psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \tag{10}$$

Para facilitar a visualização do que $(-1)^{f(x)}$ realiza, basta se atentar à equação 10. Nela, pode ser perceber que para cada x (assumindo 0 ou 1) teremos $f(x)$ sendo aplicado especificamente. Para tornar isso mais claro, vamos adotar uma versão simplificada, onde $x \in \{0,1\}^1$, podendo então representar a aplicação de U_f como

$$\frac{1}{\sqrt{2}}[(-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle] \tag{11}$$

4. Agora que já aplicamos o oráculo a todas as possíveis 2^n entradas do sistema, vamos tirar alguma conclusões sobre o que acabamos de gerar.

   Foi dito que, utilizando U_f, criamos uma interferência entre os qubits. Porém, interferências não são mensuráveis, podendo ser apenas "vistas"pela porta de Hadamard.

   Vamos novamente criar um exemplo simplificado para facilitar o entendimento do que esta sendo dito:

   Suponha que nosso $|x\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Estaremos verificando quais resultados obteremos para função constante e balanceada, ao aplicar pela segunda vez a porta de Hadamard.

   Assumindo que f é a função constante que satisfaz $f(x) = 0$, para todo o $x \in \{0,1\}$, obtemos:

$$\begin{aligned} H\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right] &= \frac{|0\rangle + |1\rangle}{2}\langle 0|0\rangle + \frac{|0\rangle - |1\rangle}{2}\langle 1|0\rangle + \\ &+ \frac{|0\rangle + |1\rangle}{2}\langle 0|1\rangle + \frac{|0\rangle - |1\rangle}{2}\langle 1|1\rangle \end{aligned}$$

   donde se conclui que

$$\begin{aligned} H\left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right] &= \frac{|0\rangle + |1\rangle}{2} + \frac{|0\rangle - |1\rangle}{2} \\ &= |0\rangle\,. \end{aligned}$$

   Perceba agora que, os dois estados $+|1\rangle$ e $-|1\rangle$ se cancelaram, enquanto $|0\rangle$ somou-se ao outro $|0\rangle$. Isso é chamado de interferência e resulta em

O mesmo ocorre para $f(x) = 1$, visto que também é constante.

Assumindo agora que f é balanceada, com $f(0) = 0$ e $f(1) = 1$:

$$\begin{aligned} H\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] &= \frac{|0\rangle + |1\rangle}{2}\langle 0|0\rangle + \frac{|0\rangle - |1\rangle}{2}\langle 1|0\rangle - \\ &- \frac{|0\rangle + |1\rangle}{2}\langle 0|1\rangle - \frac{|0\rangle - |1\rangle}{2}\langle 1|1\rangle \end{aligned}$$

ocorrendo o mesmo padrão de interferência, resulta na igualdade

$$H\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = |1\rangle\,. \tag{12}$$

Com isso temos em ψ_3 que, quando aplicamos a porta de Hadamard após termos passado as entradas pelo oráculo, a fase consegue ser distinguida (devido a capacidade de Hamadard de perceber interferência de fases) de modo que temos como saída, em ψ_3, $|0\rangle$ quando $f(x)$ é constante e $|1\rangle$ quando $f(x)$ é balanceada.

Podemos tratar tudo isso de um jeito mais inteligente, ainda utilizando a interferência. Sabemos que um escalar que esteja multiplicando um qubit é considerado sua amplitude. No caso temos $(-1)^{f(x)}\,|x\rangle$, isto é, $(-1)^{f(x)}$ é a amplitude do estado $|x\rangle$.

Sendo ψ_4 uma aplicação de Hadamard à ψ_3, tendo descartado o qubit auxiliar, temos

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)}[\sum_{y\in\{0,1\}^n}(-1)^{xy}\,|y\rangle] \tag{13}$$

que podemos escrever como

$$\frac{1}{2^n}\sum_{y\in\{0,1\}^n}[\sum_{x\in\{0,1\}^n}(-1)^{f(x)}(-1)^{xy}]\,|y\rangle \tag{14}$$

Nos interessa saber então qual a probabilidade de termos nossas saídas no valor $|0^n\rangle$, isso pois, ao que vimos para o caso simplificado, temos que nossas saídas são $|0\rangle$ quando $f(x)$ é constante.

Portanto, vamos definir $|y\rangle$ como $|000...0\rangle$:

$$\frac{1}{2^n}\sum_{y\in\{0,1\}^n}[\sum_{x\in\{0,1\}^n}(-1)^{f(x)}(-1)^{x0}]\,|000...0\rangle \tag{15}$$

Deste modo, temos

$$\frac{1}{2^n}[\sum_{x\in\{0,1\}^n}(-1)^{f(x)}]\,|000...0\rangle \tag{16}$$

Se $f(x)$ for constante, teremos

$(-1)^{f(0)} + (-1)^{f(1)} = 2^n$, i.e., eles não se cancelam.

Deste modo, a somatória para n qubits resultaria em 2^n. Teremos, portanto, nossa equação como

$$\frac{1}{2^n}2^n\,|000...0\rangle\,, \tag{17}$$

ou seja, a amplitude do estado $|000...0\rangle = 1$.

Por outro lado, caso tivessemos uma função balanceada, $f(x) \in \{0,1\}$ iria se cancelar, fazendo com que a somatória resultasse em 0 e consequentemente a probabilidade de termos $|000...0\rangle$ seria também 0.

5. E por fim, efetuamos a medição de nossos qubits, onde, como já vimos, caso tenhamos o resultado $|000...0\rangle$, será única e exclusivamente devido ao fato que nossa função é constante.

Ao estudar o algoritmo de Deutsch-Jozsa (cf. [5]), vimos como nos aproveitar de características do sistema quântico, como sobreposição e interferência, para, de maneira muito inteligente, resolver um problema de exponencialmente mais rápido do que o método clássico.

5.2 Algoritmo de Grover

Vamos agora ver outro algoritmo quântico, este com uma uso um pouco mais aplicado a computação.

Em especial, o algoritmo de Grover (cf. [10])realiza uma busca entre os 2^n estados de entrada, de modo que, para a entrada correta x, isto é, para o "item"correto, $f(x) = 1$ (cf. [33]). Ou seja, estamos a procurar por um elemento (podendo ser pensando como um elemento de um dataset) cujas características podem ser verificadas por uma função $f(x)$.

Em um análogo clássico, teríamos que aplicar no pior dos casos $f(x)N$ vezes onde N é a quantidade de itens a ser verificados. Já com o algoritmo quântico de Grover podemos fazer uso de seu paralelismo e reduzir o número de vezes que utilizamos a função $f(x)$ para $\sqrt{N}$ [15], isso pois neste aplicamos algumas vezes a função para aumentar a amplitude do estado correto.

Vamos agora nos adentrar ao funcionamento do algoritmo de Grover.

O algoritmo funciona de acordo com o diagrama abaixo:

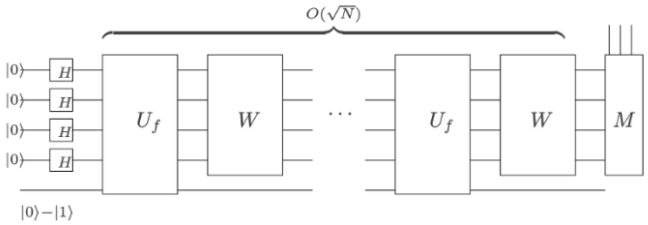

Figura 19: Circuito referente ao algoritmo de Grover.[31]

Assim como feito para o algoritmo de Deutsch-Jozsa, vamos dividir o algoritmo de Grover em passos.

1. Iniciamos o sistema em $\left|0^{\otimes n-1}\right\rangle \otimes |1\rangle$
2. Aplicamos Hadamard para cada um deles, assim cada qubit possui amplitude $\frac{1}{\sqrt{2^n}}$
3. Realizamos a iteração de Grover (aplicação de U_f e W) sob o registrador em sobreposição. Aqui mora a grande diferença em relação ao algoritmo de Deutsch-Jozsa.

   Neste, ocorrem duas operações:

   - **1º:** Ao encontrar x: $f(x) = 1$, inverte-se a amplitude do estado x. Este passo nos destaca o x que é solução do sistema
   - **2º:** Agora é realizado uma nova inversão na amplitude, porém, desta vez, a amplitude de x é invertida em torno da média. Deste modo, amplifica-se a amplitude de x.

   Este passo é repetida $\sqrt{N}$, pois a cada iteração, aumentamos a amplitude e consequentemente a probabilidade de, ao efetuar a medição, encontrarmos o estado x.
4. Efetuamos a medição.

Escrevendo os passos acima de forma mais matemática:

1. Inicia os estados em $\psi_0 = \left|0^{\otimes n-1}\right\rangle \otimes |1\rangle$
2. Aplica Hadamard a todos os vetores, sendo $|x\rangle = |000...0\rangle$:

$$\psi_1 = H^{\otimes n} \left|0^{\otimes n-1}\right\rangle |1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^{n-1}} |x\rangle |1\rangle \tag{18}$$

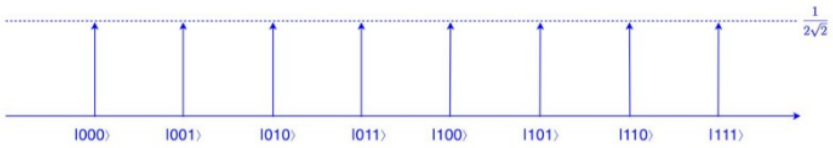

Figura 20: Distribuição de amplitudes para o sistema em sobreposição.[32]

Deste modo, temos por exemplo a distribuição de amplitudes para um sistema de 3 qubits como

3. Agora aplicamos o operador de Grover, cujo circuito pode ser visto na figura abaixo:'

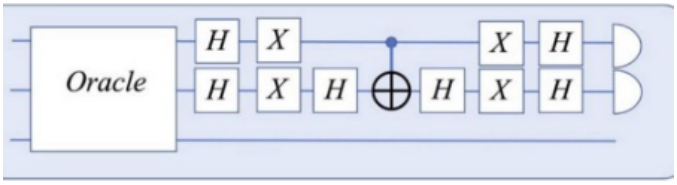

Figura 21: Operador de Grover de forma mais especifica.[33]

O operador de Grover pode ser divido em 4 etapas: O $\rightarrow H^{\otimes n} \rightarrow |0\rangle\langle 0| - I \rightarrow H^{\otimes n}$

1º: Oráculo Neste, o Oráculo é uma função do tipo

$$O : |x\rangle |y\rangle \rightarrow |x\rangle |f(x) \oplus y\rangle \tag{19}$$

onde $y = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$ e $f(x) = 1$ se x for o elemento que procuramos, caso contrário, $f(x) = 0$.

Podemos ver que, caso $f(x) = 1$, teremos

$$|x\rangle \frac{|1 \oplus 0\rangle - |1 \oplus 1\rangle}{\sqrt{2}} = - |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{20}$$

, ou seja, inverte-se $|y\rangle$, causando uma interferência em todo o sistema. Com isso, temos uma inversão na amplitude de $|x\rangle$

Para caso $f(x) = 0$, teremos

$$|x\rangle \frac{|0 \oplus 0\rangle - |0 \oplus 1\rangle}{\sqrt{2}} = |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{21}$$

[31]Disponível em https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248. Acesso em março 2019.

[32]Disponível em https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248. Acesso março 2019

[33]Disponível em https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248. Acesso março 2019

, neste caso, nem a amplitude ou o estado se altera.

Para ilustrar o caso em que $f(x) = 1$, vamos ver a imagem abaixo:

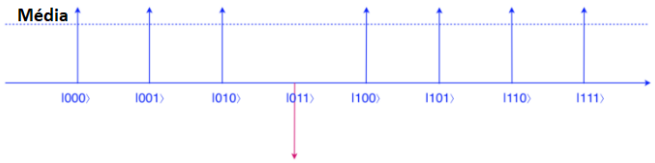

Figura 22: Inversão da fase do estado x, sendo este o estado que procurávamos.[35]

Podemos então escrever $\mathrm{O}|x\rangle\,|y\rangle = (-1)^{f(x)}\,|x\rangle$, onde descartamos o qubit auxiliar.

2º: Transformação de difusão Podemos agora aplicar o operador G qual é necessário para amplificar a amplitude de $|x\rangle$ efetuando uma inversão em torno da média.

A transformação de difusão é realizada através de outra aplicação de Hadamard nos n qubits, seguida por uma mudança de fase que inverte todos os estados, exceto $|0\rangle$ por um fator de (-1), e novamente é aplicado outra transformação de Hadamard.

Como queremos aumentar a amplitude de $|x\rangle$, vamos definir o que é a média das amplitudes:

$$\mu = \frac{1}{N} \sum_{x \in \{0,1\}^n} \alpha \tag{22}$$

Podemos então definir o difusor de grover como sendo uma função que mapeia

$$\sum_{x \in \{0,1\}} \alpha_x\,|x\rangle \rightarrow \sum_{x \in \{0,1\}} (2\mu - \alpha_x)\,|x\rangle \tag{23}$$

Ou seja, é um operador onde, caso α_x esteja abaixo da média, $|x\rangle$ tem sua amplitude aumentada. O oposto também é verdadeiro.

No caso de termos α_x uma amplitude negativa, teremos que a nova amplitude de $|x\rangle$ será $2\mu + \alpha_x$, e isto é exatamente uma reflexão em torno da média.

[35] Disponível em https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248. Acesso março 2019.

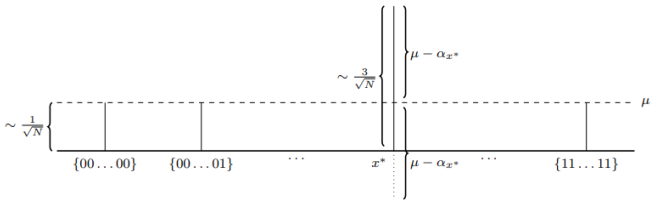

Figura 23: Inversão da fase em torno da média. Fonte: Carnegie Mellon University - repositório de aulas.[37]

Ao realizarmos continuamente, estaremos cada vez mais aumentando a amplitude do elemento que procuramos, e reduzindo as outras.

Usualmente, o difusor de Grover é escrito como $2\left|\psi\right\rangle\left\langle\psi\right| - I$, cujo ψ é ortogonal ao vetor $\left|x\right\rangle$, onde $f(x) = 1$.

Efetuamos $\sqrt{N}$ vezes a iteração de Grover devido ao fato que, a cada iteração, a amplitude α_x aumenta em um fator maior do que $\frac{1}{\sqrt{N}}$, ou seja, ao aplicarmos $\sqrt{N}$, teremos uma amplitude $\alpha_x = 1$

4. Efetua-se agora a medição, onde o estado com maior probabilidade de ser encontrado é $\left|x\right\rangle$, tal que $f(x) = 1$.

[37] Disponível em https://www.cs.cmu.edu/ odonnell/quantum15/lecture04.pdf. Acesso março 2019.

6 O algoritmo que consolidou a computação quântica

Até o momento, já vimos dois algoritmos que exemplificam bem como propriedades físicas podem ser extrapoladas criando um modelo mais eficiente de computação. No entanto, nenhuma delas foi tão importante para o campo quanto o algoritmo de Shor.

Em 1999, Peter Shor [26] criara o algoritmo que traria, de maneira definitiva visibilidade e importância aos computadores quânticos, fazendo com que mais investimentos e tempo fosse aplicado a seu estudo.

Para contextualizar e entender a importância deste algoritmo, vamos primeiro conhecer o problema que ele resolve, e onde este tipo de operação é necessária.

6.1 O problema

Na teoria dos números, o teorema fundamental da aritmética declara que qualquer número inteiro maior que 1 pode ser escrito como um produto único de números primos [2], i.e., se, por exemplo, decompormos o número 15 em seus fatores primos, teremos $5 \times 3 = 15$, sendo 5 e 3, como já sabemos, números primos. Outro detalhe deste teorema é que existe apenas uma única fatoração prima possível para cada número.

Podemos, com isso, facilmente resolver o problema de fatoração prima ao tentar dividir um dado número inteiro N por números primos, de maneira crescente, por exemplo:

$$\begin{aligned} N &= 48 \\ 48/2 &= 24 \\ 24/2 &= 12 \\ 12/2 &= 6 \\ 6/2 &= 3 \\ 3/3 &= 1 \end{aligned}$$

Ao fim, temos que o número 48 pode ser escrito pelo produto dos primos $2 \times 2 \times 2 \times 2 \times 3$.

Por mais que seja um método bem simples de obtermos a fatoração prima de um número, este torna-se inviável para números muito grandes cuja fatoração prima se da unicamente por dois números primos com uma grande quantidade de dígitos.

Fazendo uso do teorema fundamental da aritmética, em 1978 um importante sistema de criptografia foi desenvolvido por Rivest et al. (cf. [23]), criando então o chamado sistema RSA de criptografia.

O método RSA faz proveito da dificuldade em encontrar os fatores primos de um número. Neste, cria-se a chave ao pegar dois números primos grandes,

de maneira aleatória, e ao multiplica-los gera-se um número N que é tornado público.

Para encontrar a chave que descriptografa a mensagem, precisamos descobrir quais são os mesmos fatores primos que criaram o número N em questão, isto é, encontrar dois números primos p e q tal que $p \times q = N$.

Alguns algoritmos clássicos foram criados no intuito de resolver tal problema, entre eles, o mais eficiente atualmente é o General number field sieve [3] cujo tempo de execução cresce exponencialmente de acordo com o tamanho da entrada, entretanto, este ainda não é suficiente para burlar a segurança do sistema RSA, fato este que fora mudado em 1994 por Peter Shor.

6.2 Algoritmo quântico de Shor, uma intuição

Em 1994, Peter Shor publica o arquivo que traria solução potencial (dependente da real existência dos computadores quânticos) para o problema da fatoração em números primos em tempo polinomial.

Para tal, vamos com base nos trabalhos de [11], Shor reduz o problema de encontrar fatores primos a encontrar o período de uma função. Neste, devemos usar uma função periódica, ou seja, uma função cuja imagem se repete ao longo de seu domínio. Será essa a função modular:

$$f(a) = x^a \mod N$$

Podemos ver que para $a = 0$, teremos:

$$x^0 = 1$$

para qualquer x. Assim, temos

$$1 = x^0 \mod N$$

Portanto, sendo esta periôdica de período r, temos

$$1 = x^r \mod N.$$

Podemos ainda escrever

$$x^r = 1 \mod N$$
$$x^r - 1 = 0 \mod N$$

e se r for par, temos a forma expandida:

$$(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1) = x^r - 1$$

deste modo

$$(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1) = 0 mod N \tag{24}$$

Esta equação traz um sentido muito bonito e um pouco difícil de visualizar de imediato.

Sabemos que a função modulo tem como propriedade a equivalência

$$x^y \mod x = 0$$

que nos diz exatamente que x^y é um multiplo de x, portanto, não deixando restos na divisão. O mesmo pode ser observado da equação 24, no qual temos um produto que acaba sendo um múltiplo de N, deixando assim nenhum resto.

Sendo $(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1)$ um múltiplo de N, podemos rescrever nossa equação como

$$m \cdot (N) = 0 mod(N)$$

Para o caso de um número N ser o produto de dois números primos, como vimos para o método de criptografia RSA, poderemos reescrever nossa equação como

$$m \cdot (pq) = 0 mod(pq)$$

sendo m um número inteiro.

Assim, desde que tenhamos garantia de que $x \neq \pm 1$, a única solução será que o produto $(x^{\frac{r}{2}} - 1)(x^{\frac{r}{2}} + 1)$ contém fatores não triviais, sendo estes os fatores primos p e q (RSA), no qual, através do algoritmo de euclides, podemos encontrar por meio do mdc (maior divisor comum)

$$\text{mdc}(m \cdot p, pq)$$
$$\text{mdc}(m \cdot q, pq)$$

que resultará em p e q, respectivamente.

Da mesma maneira, retomando o modo que foi escrito na equação 24, teremos

$$\text{mdc}((x^{\frac{r}{2}} - 1), N)$$
$$\text{mdc}((x^{\frac{r}{2}} + 1), N)$$

Desde que x $\neq \pm 1$, o mdc acima trará no minimo um fator não trivial de N, resolvendo assim nosso problema de fatoração prima.

Antes de traduzirmos as intuições acima para o formalismo quântico e assim desenvolver um circuito para o algoritmo de Shor, vamos primeiro ver uma ferramenta muito importante para inúmeros campos do conhecimento e que é essencial para que possamos encontrar o período de nossa função.

6.3 Fourier e sua transformada quântica

As séries de Fourier e suas transformadas são ferramentas de tremenda importância para processamento de sinais, nos possibilitando realizar interpolação de dados ou de outras funções para séries de funções trigonométrica (senos e cossenos) ou até mesmo nos permitindo ter acesso ao espectro de frequências de um sinal periódico.

Vamos explorar sutilmente um pouco das séries de Fourier e as transformadas de fourier onde, ao fim, daremos enfase na transformada quântica de Fourier, ferramenta fundamental para alguns algoritmos quânticos, como é o caso do algoritmo de Shor.

6.3.1 Séries de Fourier

A série de Fourier consiste em uma interpolação de funções ou dados que apresentam periodicidade, em uma soma de senos e cossenos. É dito que todas as funções possam ser interpoladas por uma série de trigonométricas.

Podemos ver como isso funciona ao analisar a figura abaixo:

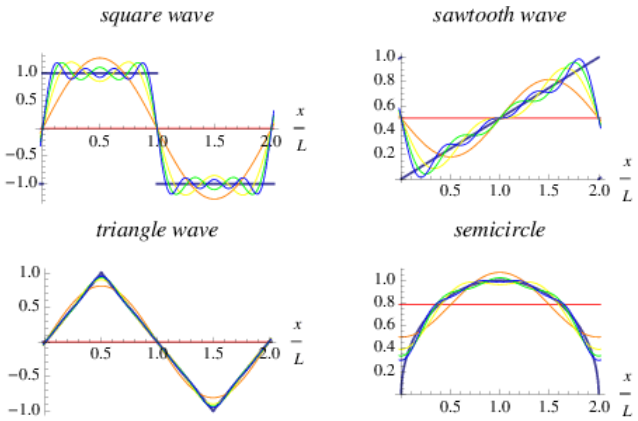

Figura 24: Uma série de funções trigonométricas somadas que se aproximam de uma função por meio da interferência.[39]

Sendo f uma função com período $r = L$, então podemos escrever a série de Fourier para tal função como [30]:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n cos\frac{n\pi x}{L} + b_n sin\frac{n\pi x}{L})$$

Entende-se na igualdade acima que, a função periódica $f(x)$, é equivalente à uma definida soma de senos e cossenos.

Os coeficientes $a_n e b_n$ são encontrados por:

[39] Disponível em: http://mathworld.wolfram.com/FourierSeries.html

$$a_n = \frac{1}{L}\int_0^L f(x)cos\frac{n\pi x}{L}dx, \quad n = 0, 1, 2, ..., \infty$$

$$b_n = \frac{1}{L}\int_0^L f(x)sin\frac{n\pi x}{L}dx, \quad n = 0, 1, 2, ..., \infty$$

Aqui, vale a pena tomar um tempo para analisar o trabalho desta integral entre duas funções, $f(x)$ e uma das trigonométricas (seno ou cosseno).

Como já vimos anteriormente em outras seções, o produto interno entre duas funções é definido com a integral destas funções, ou seja, digamos que queremos o produto interno entre $g(x)$ e $f(x)$, teremos portanto

$$\langle f(x), g(x)\rangle = \int_a^A f(x)g(x)dx$$

Sabemos que o produto interno gera um número (escalar). Por ser uma generalização do produto escalar, o seu significado geométrico é carregado, no sentido que, quando calculamos o produto interno de duas funções, estamos verificando essencialmente quão semelhante estas são. De maneira mais precisa, estamos a quantificar o quanto de uma função existe na outra. Caso as funções não contenham partes em que uma possa ser projetada na outra, então elas serão ortogonais e apresentarão $\langle f(x), g(x)\rangle = 0$.

Um caso fácil de visualizar são os dois vetores do plano cartesiano i e j que não apresentam nenhum componente em comum, sendo assim, um não pode ser projetado em outro, declarando portanto sua ortogonalidade.

Agora que já compreendemos em breve algumas das minucias que a série de Fourier carrega para que funcione como proposto, podemos tratar da mesma como uma questão de encontrar os coeficientes de Fourier (a's e b's) para que tenhamos a interpolação por series trigonométricas.

6.3.2 Série complexa de Fourier

Antes de conhecermos a transformada de Fourier, vamos primeiro criar a base para sua existência. Já temos conhecimento sobre a série de Fourier para números reais, precisamos agora verificar a série de Fourier para números complexos [29]. Entender a série complexa de Fourier é de extrema importância para que possamos apresentar a transformada de Fourier.

É quase automático que ao pensarmos em senos, cossenos e números complexos a primeira ideia que temos é a formula de Euler

$$e^{iwt} = cos(wx) + isin(wx)$$

onde i é o número complexo $\sqrt{-1}$ e x é uma variável qualquer.

Podemos, portanto, com a equação de Euler reescrever a série de Fourier:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{inwx} \tag{25}$$

onde o coeficiente c_n pode ser agora encontrado como

$$c_n = \frac{1}{T} \int_0^T f(x) e^{-inwx} \tag{26}$$

sendo T o período da função e w indica em radianos a posição no circulo trigonométrico.

6.3.3 Transformada de Fourier

Na série de Fourier vimos que podemos representar um sinal periódico por meio de uma combinação linear (soma) de senos e cossenos. Na transformada de Fourier, veremos uma extensão da série de Fourier para casos onde estamos lidando com funções não periódica.

Um jeito de extrapolarmos o que antes era definido para funções periódicas para agora onde as funções não são periódicas é declarar que a função não periódica tem um período definido entre $\pm\infty$ [18]. Este truque é bem interessante pois é como se estivéssemos assumindo que esta função contém um período, fosse periódica], entretanto está em um intervalo tão grande que é como se ela não fosse periódica.

Matematicamente, declarar que uma função tem o período definido em $\pm\infty$ traz consigo a implicação de que a frequência da função será de uma grandeza infinitesimal. Portanto, podemos evoluir as equações 25 e 26 para um período infinito, ficando:

$$c_n = \frac{1}{T} \int_0^T f(x) e^{-inwx} \rightarrow c(w)dw = \lim_{T\to\infty} \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{T}{2}} f(t) e^{-inw_0 t} \tag{27}$$

onde w_0 representa a frequência infinitesimalmente pequena. Dado o produto $n \cdot w_0$, se tivermos n sendo um valor que tende ao infinito, conseguiríamos trabalhar com algum frequência finita.

Com T indo ao infinito, temos pela fórmula que relaciona período e frequência

$$T = \frac{2\pi}{w_0} \tag{28}$$

temos

$$c(w)dw = \frac{dw}{2\pi} \int_{\infty}^{\infty} f(t) e^{-iwt} dt \tag{29}$$

onde, integrando ambos os lados com o diferencial dw

$$c(w) = \frac{1}{2\pi} \int_{\infty}^{\infty} f(t) e^{-iwt} dt \tag{30}$$

Este raciocínio nos levou a uma extrapolação do antes periódico para o não periódico, entretanto, ainda difere brevemente da transformada de Fourier $F(w)$

A correção de $c(w)$ para a transformada de Fourier $F(w)$ é dada pela igualdade $F(w) = 2\pi c(w)$. Portanto

$$F(w) = 2\pi c(w) = \frac{2\pi}{2\pi} \int_{\infty}^{\infty} f(t) e^{-iwt} dt = \int_{\infty}^{\infty} f(t) e^{-iwt} dt \tag{31}$$

Podemos também agora, ao manipular a equação acima, definir a inversa da transformada de Fourier $f(t)$

$$f(t) = \frac{1}{2\pi} \int_{\infty}^{\infty} F(w) e^{iwt} dw \tag{32}$$

.

Vale a pena agora comentar que, enquanto a transformada de Fourier realiza uma transformação do domínio do tempo $f(t)$ para o dominio da frequência $F(w)$, a inversa transforma uma função que está no domínio da frequência $F(w)$ e transforma a outra cujo domínio é o tempo $f(t)$.

6.3.4 Transformada Discreta de Fourier

A transformada de Fourier agora se expande para casos onde o sinal é conhecido apenas em instantes e não mais em um espaço contínuo como nos era fornecido por funções na Transformada de Fourier, ou seja, enquanto antes estávamos lidando com $f(t)$ sendo uma função contínua não periódica (período $\pm \infty$), agora estaremos lidando com dados discreto $f(0)$, $f(1)$, $f(2)$, ..., $f(N-1)$.

Análogo ao que fizemos na transformada de Fourier, podemos pegar um conjunto de dados discretos e dizer que este é periódico, sendo seu período definido entre o primeiro elemento do conjunto e o último. Perceba que no caso contínuo, fazíamos essencialmente a mesma coisa, no qual o intervalo era definido por -∞, ∞.

Com isso, a transformada de Fourier se transforma em sua forma discreta [24] ao realizar pequenas mudanças

$$F(w) = \int_{\infty}^{\infty} f(t) e^{-iwt} dt \rightarrow F(w) = \sum_{k=0}^{N-1} f(k) e^{-iwk} \tag{33}$$

Perceba que a frequência w é

$$w_k = \frac{2\pi}{N} k \tag{34}$$

com k indo de 0 até $N-1$.

Temos então a transformada discreta de Fourier como

$$F(n) = \sum_{k=0}^{N-1} f(k) e^{-i\frac{2\pi}{N}kn} \tag{35}$$

, onde n vai de 0 até $N-1$.

Sua forma inversa é

$$f(k) = \frac{1}{N} \sum_{n=0}^{N-1} F(n) e^{i\frac{2\pi}{N}kn} \tag{36}$$

6.3.5 Transformada quântica de Fourier

A transformada quântica de Fourier é a implementação da transformada discreta de fourier em sistemas quânticos, isto é, ao se alterar algumas notações da transformada discreta, teremos em mãos a transformada quântica [20].

Sendo a discreta definida por

$$F(n) = \sum_{k=0}^{N-1} f(k) e^{-i\frac{2\pi}{N}kn} \tag{37}$$

, a quântica realiza transformação na amplitude dos estados:

$$\sum_{j=0}^{N-1} x_j \left|j\right\rangle \rightarrow \sum_{k=0}^{N-1} y_k \left|k\right\rangle = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} x_j e^{2\pi i j \frac{k}{N}} \left|k\right\rangle \tag{38}$$

Essencialmente, conforme a equação acima, a transformada quântica recebe o sistema e transforma cada uma das amplitudes no produto $x_j e^{2\pi i j \frac{k}{N}}$, onde x_j era anteriormente a amplitude o estado.

Ao pensarmos no circulo trigonométrico complexo, podemos representar qualquer ponto no raio unitário através do termo $e^{\frac{2\pi}{N}i}$, para qualquer valor de N. Este termo é chamado de raiz da unidade, onde, para cada valor N escolhidos, teremos N soluções para a equação.

Vamos chamar $w = e^{\frac{2\pi}{N}i}$, para $N = 5$, temos as potências de w como as soluções do sistema.

Podemos facilmente visualizar isso a partir da imagem a seguir.

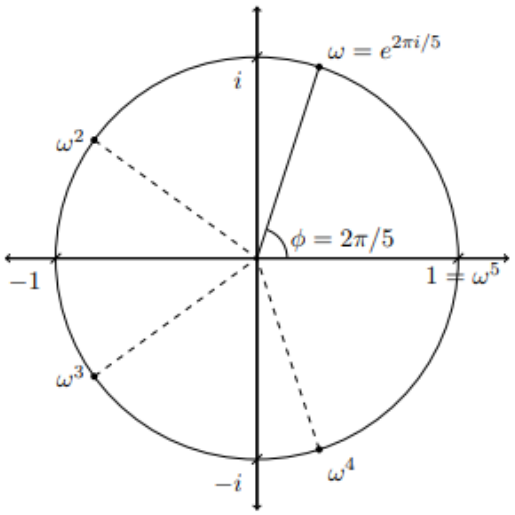

Figura 25: Círculo trigonométrico com pontos representados pelo elemento de fase.[41]

Podemos notar que cada uma das soluções apresenta certa fase ou angulo quanto à origem. Tal angulo é acessível por $\phi = \frac{2\pi}{M}$. se elevarmos w à potência j, então $\phi = \frac{2j\pi}{M}$. Assim, temos que w é um fator que carrega consigo a ideia de fase, ao operar em qubits (vetores em um espaço complexo de 3 dimensões) intuitivamente operará alterando a fase do vetor.

Tal propriedade de fase não será muito discutida aqui, entretanto, existem algoritmos que se aproveitam desta propriedade, como o algoritmo de estimação de fase.

Outro detalhe que não fora apresentado na transformada Discreta de Fourier porém é importante estar a apresentar para o análogo quântico é a sua forma matricial, isto é, a transformação Discreta assim como a quântica podem ser escritas de forma matricial como a seguir:

$$QFT_m = \frac{1}{m}\begin{Bmatrix} 1 & 1 & 1 & 1 & ... & 1 & 1 \\ 1 & w & w^2 & w^3 & w^4 & ... & w^{m-1} \\ 1 & w^2 & w^4 & w^6 & w^8 & ... & w^{2m-2} \\ 1 & w^3 & w^6 & w^9 & w^{12} & ... & w^{3m-3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{m-1} & w^{2m-2} & w^{3m-3} & w^{4m-4} & ... & w^{(m-1)(m-1)} \end{Bmatrix} \tag{39}$$

Cada elemento da matriz pode ser escrito como w^{jk}, sendo j linha e k coluna, como exemplo.

[41] Disponível em https://courses.edx.org/c4x/BerkeleyX/CS191x/asset/chap5.pdf

Para concluir, vale enfatizar/relembrar que a transformada quântica de Fourier, assim como a transformada discreta, tem como domínio a frequência.

6.4 Algoritmo quântico de Shor. Implementação

Vamos agora passar todas as intuições anteriores para o cenário concreto da computação quântica.

Em resumo, estamos interessados em encontrar o período da função $f(a) = x^a \mod N$ e a partir disso, calcular o $\text{mdc}(x^{\frac{r}{2}} - 1, N)$ e $\text{mdc}(x^{\frac{r}{2}} + 1, N)$, nos permitindo assim obter os fatores primos p e q tal que $p \cdot q = N$.

Para tal, vamos apresentar por tópicos o passo a passo para construir o circuito referente ao algoritmo de Shor:

- Antes de mais nada, precisamos definir o valor de x, referente a função $f(a) = x^a \mod N$. O x pode ser escolhido aleatoriamente, desde que seja coprimo de N, isto é, $\text{mdc}(x, N) = 1$. Tal procedimento pode ser realizado em um computador clássico, utilizando o próprio algoritmo de Euclid (mdc).

- Agora, vamos construir nosso registrador de uma forma um pouco diferente das que fizemos em outros algoritmos. Estaremos dividindo-o em duas partes. Em cada uma das duas teremos n qubits, onde n é a quantidade de bits necessário para representar m números. Isto é, caso m seja o número 1020, como exemplo, precisaremos de 10 qubits, pois $2^{10} = 1024$, o que nos permite representar o intervalo de números inteiros de 0 até 1023. O número m é definido pelo intervalo $N^2 \leq m \leq 2N^2$.

- Para o primeiro registrador, aplicaremos a porta de Hadamard

$$H^{\otimes n} |0\rangle \tag{40}$$

  Este primeiro registrador em sobreposição representará simultaneamente todos os números entre 0 até $m - 1$, sendo estes os valores de a, para a função $f(a) = x^a \mod N$.

- O segundo registrador será iniciado com todos os n qubits no estado $|0\rangle$, portanto

$$|0\rangle^{\otimes n} \tag{41}$$

- Agora, construiremos um oráculo que carregue a função $f(a) = x^a \mod N$. Em seguida, estaremos aproveitando o paralelismo quântico e alimentando este oráculo com a sobreposição criada no primeiro registrador. Assim, estaremos calculando $x^a \mod N$ para todos os $m-1$ números representados como a em uma única iteração.

Após aplicar o oraculo, salvemos os resultados no segundo registrador, mantendo o primeiro no mesmo estado que antes de aplicar o oraculo.

Assim, temos nosso sistema no estado

$$\frac{1}{\sqrt{m}} \sum_{a=0}^{m-1} |a, x^a mod N\rangle \tag{42}$$

Observe que na equação acima, a virgula dentro do ket declara que são registradores distintos.

- Neste momento, o segundo registrador estará em uma sobreposição com todos os resultados da operação anterior, onde todos os resultados apresentam a mesma amplitude (raiz da probabilidade). Ao realizarmos uma medição no segundo registrador, estaremos colapsando o sistema para um número "k"aleatório (um dos quaisquer valores em sobreposição). No mesmo instante que fazemos isso, o nosso primeiro registrador sofre uma mudança, indo de uma sobreposição de $m-1$ números para apenas aqueles números que, quando colocados na função $f(a) = x^a \mod N$, resultam em k, ou seja, para valores de a tal que $k = x^a \mod N$.

  Assim, nosso sistema se torna uma sobreposição de a's que formam o conjunto imagem de $f(a) = k$, estando k contido no segundo registrador:

$$\frac{1}{\sqrt{|A|}} \sum_{a':a' \epsilon A} \left|a', k\right\rangle \tag{43}$$

  Sendo A o conjunto de números que quando substituídos em a, resulta em $f(a) = k$.

- Retomando a ideia de que $x^a \mod N$ é periódica e portanto, dado um período r temos $x^a \mod N = x^{a+r} \mod N$, podemos dizer que os valores de a contidos no conjunto A são justamente valores de a separados por um múltiplo do período, ou seja, temos nosso registrador 1 no estado

$$|x^a\rangle \left|x^{a+r}\right\rangle \left|x^{a+2r}\right\rangle \left|x^{a+3r}\right\rangle ... \tag{44}$$

  Graficamente, podemos exemplificar o estado do nosso registrador como

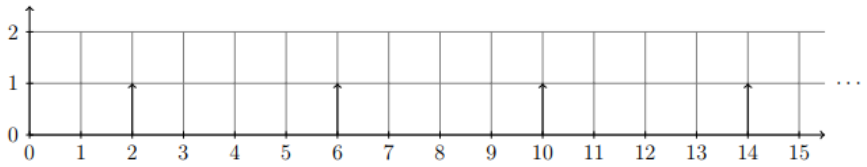

Figura 26: Estado resultante ao aplicar a transformada quântica de Fourier.[43]

- Podemos agora aplicar a transformada quântica de Fourier ao registrador 1.

$$\frac{1}{\sqrt{|A|}} \sum_{a \in A} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{\frac{2\pi}{q} a'c} \, |c, k\rangle$$

Este é um truque muito interessante. Vamos olhar com calma o que está acontecendo aqui.

Tínhamos uma informação bem local (específica) gerada exclusivamente por um valor k, que nos dava valores distanciados por período r, do tipo $x_0 + c \cdot r$. Poderíamos pensar em obter este fator r ao comparar dois ou mais valores desta distribuição periódica, entretanto, como sabemos, ao realizarmos uma medida do sistema, ocorrerá um colapso e todas as informações que não a medida serão perdidas.

Uma solução para este impasse seria realizar a medição, como falado anteriormente, reconstruir o estado e realizar novamente uma medição. Entretanto, como $f(a)$ pode conter distintos valores possíveis para k, a informação levantada fazendo este processo pode nos levar a lugar algum, ou ainda pior, nos levar a conclusões erradas.

Eis a importância de aplicar a transformada de Fourier. Independente do valor colapsado k, todos os conjuntos de a's soluções para $f(a) = k$ serão periódicos e os períodos serão exatamente iguais. Portanto, independente do k, a equação que relaciona a frequência fundamental (w) com o tamanho do vetor (M) e o período (r):

$$w = \frac{M}{r} \tag{45}$$

será igual para todos os possíveis sistemas obtidos com qualquer um dos k's.

Como já vimos, quando aplicamos a transformada de Fourier, estamos transformando nossa função (com domínio sendo dados discretos

[43] Disponível em https://www.cs.cmu.edu/~odonnell/quantum15/lecture08.pdf

de a's) para um coeficiente com caráter de soma trigonométrica cujo domínio é o das frequências.

Por se tratar de uma sequência periódica de valores a's, o espectro da frequência, gerado pela transformada de Fourier, irá conter múltiplos da **frequência fundamental**. Pela equação 45, temos que a frequência fundamental é um fator mais global, por assim dizer.

Com isso, podemos em suma dizer que, ao aplicar a transformada de Fourier a um conjunto de a's, independente do k referente ao conjunto dos a's, conseguimos trazer um termo em comum que é a frequência fundamental, nos permitindo assim realizar a comparação como mencionado anteriormente.

- Efetuamos agora a medição do registrador 1, que ira resultar em um múltiplo da frequência fundamental e, portanto, em um múltiplo de $\frac{M}{r}$.
- Guarde o valor encontrado no passo anterior e repita todo o processo. Ao fim, teremos valores múltiplos da frequência fundamental. Por fim, podemos calcular o mdc destes múltiplos de $\frac{M}{r}$, o que vai resultar em $\frac{M}{r}$. Como sabemos o valor de M (vetor de entradas em sobreposição, isto é 2^n, sendo n a quantidade de qubits do registrador 1), então podemos facilmente calcular o valor de r. Caso r seja um valor ímpar, devemos descartá-lo, escolher outro valor para x e rodar novamente o algoritmo.
- Agora que já temos o valor de r, podemos usar novamente o algoritmo de Euclid para calcular

$$\text{mdc}(x^{\frac{r}{2}} - 1, N)$$
$$\text{mdc}(x^{\frac{r}{2}} + 1, N)$$

sendo o resultado destes os fatores primos p e q tal que $p \cdot q = N$.

O algoritmo de Shor (cf. [26]), tem em sua parte quântica, o circuito na seguinte forma:

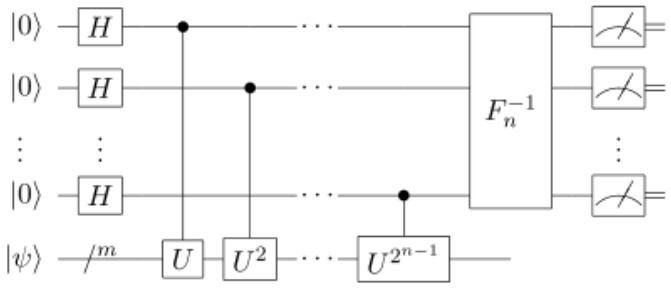

Figura 27: Circuito quântico referente ao algoritmo de Shor.[44]

[44]Disponível em https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Quantum_

7 Posfácio

Ao fim deste trabalho, algumas considerações são cabíveis e importantes de se destacar. A começar por dizer que os estudos aqui realizados, mais precisamente o estudo da Álgebra Linear e da Física Quântica, não somente nos permite acesso sobre a verdade da nossa natureza microscópica mas também nos mune de ferramentas poderosíssimas na criação de ideias e soluções de problemas servindo-nos também ao nos treinar a lidar com ideias complexas e que fogem à intuição.

No capítulo 4 vimos como podemos representar elementos comuns à computação através de objetos que pertencem ao campo de estudo da física quântica, fazendo deste um exemplo bem claro e importante da interdisciplinaridade. Ainda neste, tivemos a apresentação do qubit como uma unidade elementar de informação, qual é responsável por fundamentar todo o potencial da computação quântica, devido as características de sobreposição e interferência, conceitos estes explorados por meio dos algoritmos quânticos.

Estando familiarizado com os elementos que compõem um circuito quântico, iniciamos no capítulo 5 o estudo dos algoritmos quânticos. Neste, tanto o algoritmo de Deutsch-Josza quanto o algoritmo de Grover, nos permitiram visualizar a aplicação das propriedades quântica, tal como o paralelismo e a interferência. O algoritmo de Grover, ainda mais sofisticado do que o algoritmo de Deutsch-Josza, nos mostra como contornar e reduzir a chance de erro eminente, consequente do colapso da função de onda, por fazer inversões na amplitude.

Para completar, o algoritmo de Shor tratado no capítulo 6 vem para firmar o potencial que o computador quântico tem. Neste, ele demonstra que resolver o problema de fatoração em números primos é uma tarefa que pode ser feita em tempo útil suficiente para quebrar o sistema de criptografia RSA.

Por fim, conforme era proposto a este trabalho, podemos concluir que o objetivo de se introduzir ao campo da computação quântica foi cumprido, restando, portanto, uma continuação nos estudos para melhor apreciação e aprofundamento das possibilidades aqui apresentadas.

Referências

[1] Cao, Z., Uhlmann, J., & Liu, L. (2018). Analysis of Deutsch-Jozsa Quantum Algorithm. IACR Cryptology ePrint Archive, 2018, 249. {26}

[2] Carl, F. (1986). *Gauss. Disquisitiones Arithmeticae. G. Fleischer, Leipzig, 1801.* English translation by AA Clarke. {35}

[3] Case, M. (2003). *A beginner's guide to the general number field sieve.* Oregon State University, ECE575 Data Security and Cryptography Project. {36}

[4] de Lima Marquezino, F., Portugal, R., & Lavor, C (2019). *A Primer on Quantum Computing*, Springer Nature Switzerland AG. {3,11,14}

[5] Deutsch, D., & Jozsa, R. (dezembro de 1992). *Rapid solution of problems by quantum computation.* Em Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences (Vol. 439, No. 1907, pp. 553-558). The Royal Society. {3,25,27,30}

[6] Dirac, P. A. M. (1939, July). *A new notation for quantum mechanics.* In Mathematical Proceedings of the Cambridge Philosophical Society (Vol. 35, No. 3, pp. 416-418). Cambridge University Press. {6,11,18}

[7] Feynman, R. P. (1982). *Simulating physics with computers.* International journal of theoretical physics, 21(6), 467-488. {3,4}

[8] Glendinning, I. (2005, February). *The bloch sphere.* In QIA Meeting TechGate. {12}

[9] Griffiths, D. J., & Schroeter, D. F. (2018). *Introduction to quantum mechanics.* Cambridge University Press. {6,11}

[10] Grover, L. K. (julho de 1996). *A fast quantum mechanical algorithm for database search.* In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (pp. 212-219). ACM. {3,30}

[11] Hayward, M. *Quantum Computing and Shor's Algorithm.* Retrieved December 01, 2016. {36}

[12] Hui, J. A Medium Corporation, 2018. Disponível em: https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-8996b667d256. Acesso em: fev. 2019 {14}

[13] Hui, J. A Medium Corporation, 2018. Disponível em: https://medium.com/@jonathan_hui/qc-programming-with-quantum-gates-2-qubit-operator-871528d136db. Acesso em: fev. 2019 {19}

[14] Hui, J. A Medium Corporation, 2018. Disponível em: https://medium.com/@jonathan_hui/qc-quantum-algorithm-with-an-example-cf22c0b1ec31. Acesso em: mar. 2019 {26}

[15] Hui, J. A Medium Corporation, 2018. Disponível em: https://medium.com/@jonathan_hui/qc-grovers-algorithm-cd81e61cf248. Acesso em: mar. 2018 {30}

[16] Kaye, P., Laflamme, R., & Mosca, M. (2007). *An introduction to quantum computing*. Oxford University Press. {11,14,18}

[17] Moore, G. E. (1998). *Cramming more components onto integrated circuits*. Proceedings of the IEEE, 86(1), 82-85. {4}

[18] Murray, D. (2011). *From Complex Fourier Series to Fourier Transforms*. Department of Information Engineering, University of Oxford. https://www.robots.ox.ac.uk/ dwm/Courses/2TF_2011/2TF-N2.pdf Acesso em: jun. 2019 {40}

[19] Nambiar, Raghu & Poess, Meikel. (2010). *Transaction Performance vs. Moore's Law: A Trend Analysis*. 110-120. 10.1007/978-3-642-18206-8_9. {4}

[20] *QFT, Period Finding & Shor's Algorithm* https://courses.edx.org/c4x/BerkeleyX/CS191x/asset/chap5.pdf Acesso em: jun. 2019 {42}

[21] *Quantum Theory of Radiation Interactions*. Fall 2012. Massachusetts Institute of Technology: MIT OpenCouseWare, https://ocw.mit.edu/courses/nuclear-engineering/22-51-quantum-theory-of-radiation-interactions-fall-2012/lecture-notes/MIT22_51F12_Ch2.pdf. License: Creative Commons BY-NC-SA. {6}

[22] Repositório virtual da Faculdade de Física, Universidade Estadual de São Petersburgo. *Lecture 3: Operators in Quantum Mechanics*. http://www.phys.spbu.ru/content/File/Library/studentlectures/schlippe/qm07-03.pdf. Acesso em: fev. 2019. {6}

[23] Rivest, R. L., Shamir, A., & Adleman, L. (1978). *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2), 120-126. {4,35}

[24] Roberts, S. *Lecture 7 - The Discrete Fourier Transform*. Department of Information Engineering, University of Oxford. http://www.robots.ox.ac.uk/ sjrob/Teaching/SP/l7.pdf Acesso em: jun. 2019 {41}

[25] Schroers, B. J. (2007-2008). *Module F14ZD1: Quantum Computing*. Notas de aula no website de Heriot Watt University. http://www.macs.hw.ac.uk/ bernd/F14ZD1/qcnotes.pdf. Acesso em: fev. 2019 {19}

[26] Shor, P. W. (1999). *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. SIAM review, 41(2), 303-332. {3,4,35,47}

[27] Simonite, T. (2016). *Moore's Law Is Dead. Now What?* MIT Tecnology Review. https://www.technologyreview.com/s/601441/moores-law-is-dead-now-what/. Acesso em: jun. 2018 {4}

[28] Strubell, E. (2011). *An introduction to quantum algorithms*. COS498 Chawathe Spring, 13, 19. {14,19}

[29] TheFourierTransform.com, 2010. Disponível em: http://www.thefouriertransform.com/series/complexcoefficients.php. Acesso em: fev. 2019 {39}

[30] Tseng, Z. S. *Second Order Linear Partial Differential Equations - Part II*. Department of Mathematics, Pennsylvania State University. https://www.math.psu.edu/tseng/class/Math251/Notes-PDE%20pt2.pdf Acesso em: jun. 2019. {38}

[31] Van Gael, J. (2005). *The Role of Interference and Entanglement in Quantum Computing*. {24}

[32] WATROUS, John. (2006). *Lecture 5: A simple searching algorithm; the Deutsch-Jozsa algorithm*. University of Calgary. https://cs.uwaterloo.ca/ watrous/LectureNotes/CPSC519.Winter2006/05.pdf. Acesso em: fev. 2019. {25,26}

[33] WRIGHT, John. (2015). *Lecture 4: Grover's Algorithm*. Carnegie Mellon University. https://www.cs.cmu.edu/ odonnell/quantum15/lecture04.pdf. Acesso em: fev. 2019. {30}

[34] Yirka, B. (2019). *IBM announces that its System Q One quantum computer has reached its 'highest quantum volume to date'*. Phys.org. https://phys.org/news/2019-03-ibm-quantum-highest-volume-date.html Acesso em: jun. 2018 {5}

[35] Zwiebach, B. (2013). *MULTIPARTICLE STATES AND TENSOR PRODUCTS*. {24}

—