Notas em Matemática Aplicada 8

Editado por

Eliana X.L. de Andrade
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Rubens Sampaio
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro, RJ, Brasil

Geraldo N. Silva
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Sociedade Brasileira de Matemática Aplicada e Computacional

Notas em Matemática Aplicada

1. Restauração de Imagens com Aplicações em Biologia e Engenharia

   Geraldo Cidade, Antônio Silva Neto e Nilson Costa Roberty

2. Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos

   Leandro dos Santos Coelho

3. Modelos Matemáticos e Métodos Numéricos em Águas Subterrâneas

   Edson Wendlander

4. Métodos Numéricos para Equações Diferenciais Parciais

   Maria Cristina de Castro Cunha e Maria Amélia Novais Schleicher

5. Modelagem em Biomatematica

   Joyce da Silva Bevilacqua, Marat Rafikov e Cláudia de Lello Courtouke Guedes

6. Métodos de Otimização Randômica: algoritmos genéticos e "simulated annealing"

   Sezimária F. Pereira Saramago

7. "Matemática Aplicada à Fisiologia e Epidemiologia"

   H.M. Yang, R. Sampaio e A. Sri Ranga

UMA INTRODUÇÃO À COMPUTAÇÃO QUÂNTICA

Renato Portugal - LNCC
portugal@lncc.br

Carlile Campos Lavor - UERJ
carlile@ime.uerj.br

Luiz Mariano Carvalho - UERJ
luizmc@ime.uerj.br

Nelson Maculan - UFRJ
maculan@cos.ufrj.br

Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil
2004

Coordenação Editorial: Véra Lucia da Rocha Lopes

Coordenação Editorial da Série: Geraldo Nunes Silva

Editora: SBMAC

Impresso na Gráfica: Epecê Gráfica

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Catalogação elaborada pela Biblioteca do IMECC/UNICAMP.

Portugal, Renato
Uma Introdução à Computação Quântica - São Carlos, SP : SBMAC, 2004
ix, 62 p. - (Notas em Matemática Aplicada; 8)

ISBN 85-86883-17-4

1. Computação Quântica. 2. Algoritmos Quânticos. 3. Circuitos Quânticos. I. Portugal, Renato. II. Lavor, Carlile Campos. III. Carvalho, Luiz Mariano. IV. Maculan, Nelson. V. Título. VI. Série

CDD - 530.12

Prefácio

Apresentamos um estudo introdutório à computação quântica. Esse é um domínio recente que combina três áreas bem conhecidas: matemática, física e computação.

Vamos nos concentrar em aspectos matemáticos da computação quântica. Apesar de desejável, nenhum conhecimento prévio sobre física ou computação é necessário. Quanto à matemática, a principal exigência é um curso básico de álgebra linear.

O texto está dividido em quatro capítulos. No Capítulo 1, fazemos uma breve exposição sobre computadores clássicos (Seção 1.1) e apresentamos os conceitos básicos usados no texto (Seção 1.2). Comparamos, rapidamente, computadores clássicos e quânticos na Seção 1.1 (essa discussão será mais útil para aqueles com algum conhecimento de computação). A Seção 1.2 é fundamental para todo o livro e deverá ser consultada constantemente.

No Capítulo 2, descrevemos alguns dos circuitos quânticos que serão utilizados nos capítulos seguintes. Nos Capítulos 3 e 4, cremos, está a nossa principal contribuição: produzir um texto em português que estimule o estudante de graduação, em qualquer área de ciências exatas, a estudar o assunto. Nesses capítulos, descrevemos os dois algoritmos mais divulgados em computação quântica: o algoritmo de Grover (Capítulo 3) e o algoritmo de Shor (Capítulo 4). O quarto capítulo é denso e, por isso, exigirá uma leitura mais atenta. No entanto, o texto tem todas as definições e referências necessárias para a compreensão desse algoritmo fundamental.

Existem ótimos livros sobre o assunto em língua inglesa (veja a bibliografia). O mais famoso, já um clássico, é o livro de Michael A. Nielsen e Isaac L. Chuang [16]. Uma tradução para a língua portuguesa está sendo concluída pelo Prof. Ivan dos Santos Oliveira Júnior, do Centro Brasileiro de Pesquisas Físicas (CBPF).

Para futuras edições melhoradas de nosso trabalho, gostaríamos de receber críticas e sugestões por parte dos leitores.

Finalmente, agradecemos o apoio da Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), do Programa Institutos do Milênio (Informação Quântica), da FAPERJ, do CNPq e, em particular, ao Prof. Rubens Sampaio, pelo incentivo.

Os Autores

Rio de Janeiro, 21 de junho de 2004.

Conteúdo

Capítulo 1

Conceitos Básicos

1.1 O Computador Clássico

Um computador clássico pode ser descrito de forma bastante genérica como uma máquina que lê um certo conjunto de dados, codificado em zeros e uns, executa cálculos e gera uma saída também codificada em zeros e uns. Zeros e uns são estados que podem ser representados fisicamente. No caso dos computadores clássicos, através do potencial elétrico: 0 é um estado de baixo potencial elétrico e 1 é um estado de alto potencial elétrico.

Zeros e uns formam um número binário que pode ser convertido para a base decimal. Pensemos, então, num computador como um dispositivo que calcula uma função $f : \{0, \ldots, N-1\} \to \{0, \ldots, N-1\}$, onde $N = 2^n$ (n é o número de bits usados na memória do computador). Sem perda de generalidade, consideremos que o domínio e a imagem de f são do mesmo tamanho. A cada conjunto de n bits de entrada, corresponde um único conjunto de n bits de saída, o que caracteriza f como uma função. Representamos o processo de cálculo na Figura 1.1, onde à esquerda, temos os bits de entrada e à direita, os de saída (o processo de cálculo ocorre da esquerda para a direita).

Em geral, f é descrita por blocos elementares que podem ser implementados fisicamente por transistores e outros componentes eletrônicos. Os blocos são as portas lógicas AND, OR e NOT, conhecidas como portas universais (na verdade, basta apenas a porta NOT e uma das duas outras portas, OR ou AND). Por exemplo, um exemplo de circuito que realiza a soma em aritmética módulo 2 de dois números, cada um com um bit, é apresentado na Figura 1.2. As entradas possíveis são 00, 01, 10 ou 11. As entradas são produzidas através de diferenças de potencial elétrico que geram corrente elétrica. Por sua vez, a corrente se propaga através dos fios, da esquerda para a direita, ativando as portas lógicas. Os símbolos de medida, à direita, representam que medidas de corrente são realizadas, indicando o valor de cada bit: 0 ou 1. O bit, na posição inferior, dá o resultado da operação. O fio para o bit da posição superior é desnecessário, sendo utilizado apenas para exibir a mesma quantidade de bits de entrada e saída.

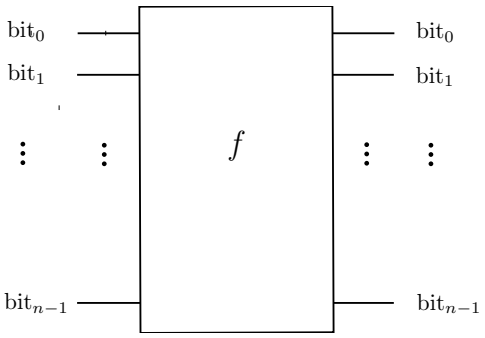

Figura 1.1: Esquema genérico para um computador clássico.

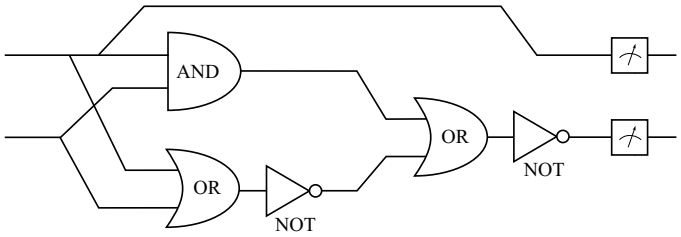

Figura 1.2: Circuito para realizar a soma de dois números, em aritmética módulo 2, cada um com um bit.

O circuito da Figura 1.2 é irreversível, pois as portas AND e OR são irreversíveis. Isso significa, no caso da porta AND, que se a saída for 0, não se sabe quais os valores dos dois bits de entrada. Para a porta OR, ocorre o mesmo, caso a saída seja 1. As portas AND e OR, descritas dessa forma, não podem ser representadas por portas quânticas, pois como veremos, são reversíveis.

No entanto, o circuito apresentado na Figura 1.2 pode ser transformado em um equivalente reversível. Para tanto, vamos utilizar a porta CNOT, representada na Figura 1.3. O valor do bit superior (chamado bit de controle) nunca muda nessa porta, enquanto que o bit inferior (chamado bit alvo) é alterado apenas se $a = 1$. Se $a = 0$, nada acontece a ambos os bits (no caso quântico, que será visto adiante, o comportamento é bem diferente). A porta CNOT é uma porta NOT, controlada pelo valor do bit superior. Podemos verificar que o valor do bit inferior de saída é dado por $a + b \pmod 2$.

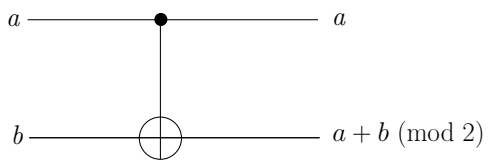

Figura 1.3: Porta CNOT.

Generalizando a porta CNOT, usando dois bits de controle no lugar de apenas um, temos a porta Toffoli (Figura 1.4), que pode ser usada para obter a contrapartida reversível da porta AND.

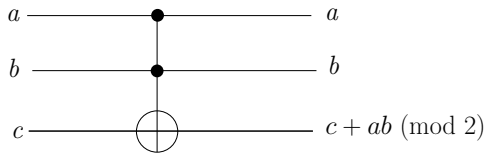

Figura 1.4: Porta Toffoli.

O valor do bit inferior (o bit alvo) é invertido apenas se a e b valem 1. Caso contrário, nada é alterado. A seguir, descrevemos todas as possíveis entradas e as saídas correspondentes:

$$\begin{aligned} 000 &\rightarrow 000 \\ 001 &\rightarrow 001 \\ 010 &\rightarrow 010 \\ 011 &\rightarrow 011 \\ 100 &\rightarrow 100 \\ 101 &\rightarrow 101 \\ 110 &\rightarrow 111 \\ 111 &\rightarrow 110 \end{aligned}$$

A porta AND pode ser representada por uma porta Toffoli colocando $c = 0$. A saída do bit inferior será, então, a AND b. Para obter o equivalente reversível para a porta OR, consulte [16].

Ainda na Figura 1.2, observe que há uma bifurcação de fios e não há problema algum em fazê-lo classicamente. Entretanto, isso não é possível em circuitos quânticos, devido ao teorema de "não clonagem" (veja [19], p. 162). Verifique que esse efeito pode ser obtido através de uma porta CNOT, colocando $b = 0$. Com isso, o valor do bit superior será duplicado.

Consideremos, novamente, a Figura 1.1. Se o computador tem n bits de entrada, há 2^n entradas possíveis, e, para cada uma delas, há também 2^n saídas possíveis.

Com isso, o número de funções que pode ser obtido é $(2^n)^{2^n}$, ou seja, 2^{n2^n}. Todas essas funções podem ser reduzidas a circuitos usando as portas universais [16, 18].

Uma questão fundamental é a "velocidade" com que um computador calcula essas funções. Isso dependerá do número de portas usadas no circuito que calcula f. Se o número de portas cresce polinomialmente com n, dizemos que o circuito é eficiente. Por outro lado, se o número de portas cresce exponencialmente com n, dizemos que o circuito é ineficiente. Esse é um método grosseiro de medida de eficiência, mas útil para a análise teórica quando n é grande.

Todos os cálculos realizados em um computador clássico também podem ser efetuados em computadores quânticos. Basta substituirmos as portas irreversíveis clássicas pelas homólogas reversíveis quânticas. Entretanto, o atrativo da computação quântica é a possibilidade de se ter algoritmos quânticos mais rápidos que os clássicos, para uma mesma classe de problemas. Para tanto, os algoritmos quânticos devem usar propriedades quânticas, não disponíveis nos computadores clássicos, como o paralelismo quântico e o emaranhamento.

1.2 O Computador Quântico

1.2.1 O bit quântico (q-bit)

Em computação quântica, utilizam-se estados quânticos em vez de estados clássicos. O bit é, então, substituído pelo bit quântico, o *q-bit*, e os valores 0 e 1 de um bit são substituídos pelos vetores $|0\rangle$ e $|1\rangle$, representados por

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Essa notação, utilizada em mecânica quântica, é conhecida por notação de Dirac.

A diferença entre um bit e um q-bit é que um q-bit genérico $|\psi\rangle$ pode também ser uma combinação linear dos vetores $|0\rangle$ e $|1\rangle$, ou seja,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.1}$$

onde α e β são números complexos. Note que os vetores $|0\rangle$ e $|1\rangle$ formam uma base ortonormal do espaço vetorial $\mathbb{C}^2$. Essa base é chamada de *base computacional* e o vetor $|\psi\rangle$ é chamado de *superposição* dos vetores $|0\rangle$ e $|1\rangle$, com *amplitudes* α e β. Em mecânica quântica, vetor é também chamado de *estado*. Usaremos os dois termos com o mesmo significado.

A interpretação física do q-bit, em (1.1), é que ele está simultaneamente nos estados $|0\rangle$ e $|1\rangle$. Isso faz com que a quantidade de informação que pode ser armazenada no estado $|\psi\rangle$ seja infinita. Entretanto, essa informação está no nível quântico. Para torná-la acessível, no nível clássico, precisamos fazer uma medida. A mecânica quântica diz que o processo de medida altera o estado de um q-bit, fazendo-o assumir o estado $|0\rangle$, com probabilidade $|\alpha|^2$, ou o estado $|1\rangle$, com probabilidade $|\beta|^2$ (isso significa que os valores α e β não podem ser conhecidos através

de uma medida). Com apenas duas possibilidades, $|0\rangle$ ou $|1\rangle$, temos, então,

$$|\alpha|^2 + |\beta|^2 = 1. \tag{1.2}$$

Isso significa que a norma do vetor $|\psi\rangle$ vale 1 (vetor unitário). Resumindo: matematicamente, um q-bit é um vetor de norma 1 de $\mathbb{C}^2$.

Na verdade, a definição da base computacional deveria ser

$$|0\rangle = \begin{bmatrix} (1,0) \\ (0,0) \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} (0,0) \\ (1,0) \end{bmatrix},$$

pois todas as coordenadas são números complexos. Para simplificar a notação, usaremos 1 para representar (1,0) e 0 para representar (0,0).

Na equação (1.2), considere $\alpha = a + i\,b$ $(a, b \in \mathbb{R})$ e $\beta = c + i\,d$ $(c, d \in \mathbb{R})$. Como $|\alpha|^2 = (\sqrt{a^2 + b^2})^2$ e $|\beta|^2 = (\sqrt{c^2 + d^2})^2$, podemos escrever

$$a^2 + b^2 + c^2 + d^2 = 1. \tag{1.3}$$

Nesse caso, podemos interpretar um q-bit como sendo um vetor unitário de $\mathbb{R}^4$. Entretanto, existe uma representação geométrica de um q-bit em $\mathbb{R}^3$: a *esfera de Bloch* (Figura 1.5). Para tanto, passemos o q-bit

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{1.4}$$

de coordenadas cartesianas para coordenadas polares (como anteriormente, $\alpha = a + i\,b$ e $\beta = c + i\,d$ $(a, b, c, d \in \mathbb{R})$). Usando as representações polares de α e β,

$$\alpha = |\alpha| \exp(i\gamma) \qquad \text{e} \qquad \beta = |\beta| \exp(i(\gamma + \varphi)),$$

e definindo

$$\cos(\theta/2) = |\alpha| \qquad \text{e} \qquad \operatorname{sen}(\theta/2) = |\beta|,$$

ou ainda

$$\begin{aligned} \theta &= 2 \arccos(\sqrt{a^2 + b^2}) = 2 \operatorname{arcsen}(\sqrt{c^2 + d^2}), \\ \varphi &= \arg(\beta) - \arg(\alpha), \\ \gamma &= \arg(\alpha), \end{aligned} \tag{1.5}$$

podemos, finalmente, escrever

$$|\psi\rangle = \exp(i\gamma)[\cos(\theta/2)\ |0\rangle + \exp(i\varphi) \operatorname{sen}(\theta/2)\ |1\rangle]. \tag{1.6}$$

EXERCÍCIO 1.1 Usando as definições dadas em (1.5), demonstre que a expressão (1.4) pode ser escrita na forma (1.6).

Para fins de representação, vamos desconsiderar o termo externo aos colchetes, $\exp(i\gamma)$, também chamado *fator de fase global*. Uma razão que permite essa simplificação é que o valor do quadrado do módulo das amplitudes de um q-bit não se altera, quando excluímos esse fator. Por exemplo:

$$|\alpha|^2 = |\exp(i\gamma)\cos(\theta/2)|^2 = |\exp(i\gamma)|^2|\cos(\theta/2)|^2 = |\cos(\theta/2)|^2,$$

o mesmo ocorrendo com $|\beta|^2$ (para um tratamento detalhado desse fato, consulte [16], p. 93). Ficamos, então, com uma representação de três parâmetros: dois explícitos, θ e φ, e um implícito, o comprimento do vetor, que é sempre igual a um. Esses parâmetros podem ser utilizados para obtermos uma representação polar no $\mathbb{R}^3$, da forma

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\varphi \operatorname{sen}\theta \\ \operatorname{sen}\varphi \operatorname{sen}\theta \\ \cos\theta \end{bmatrix},$$

onde $0 \leq \theta \leq \pi$ e $0 \leq \varphi < 2\pi$.

Usando essas convenções, a representação da base computacional, na esfera de Bloch (Figura 1.5), será:

$$|0\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}.$$

Ou seja, $|0\rangle$ será o pólo norte da esfera e $|1\rangle$ será seu pólo sul.

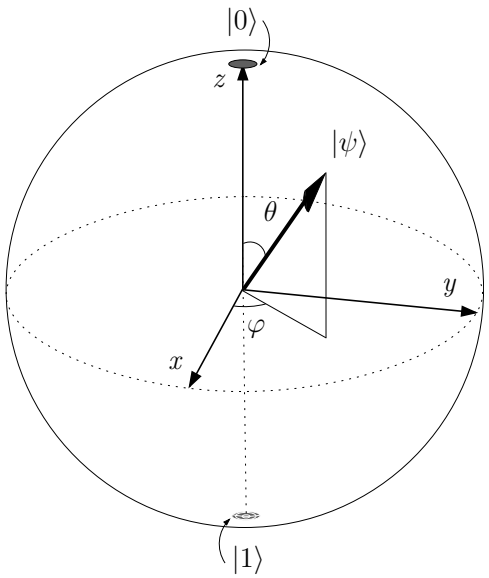

Figura 1.5: Esfera de Bloch.

Dessa forma, todos os estados de um q-bit podem ser representados (a menos de um fator multiplicativo) na esfera de Bloch. Por exemplo, os estados $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, que serão utilizados mais à frente, são representados por (1, 0, 0) e (-1, 0, 0), respectivamente.

Exercício 1.2 Dê uma interpretação, em termos de amplitudes e probabilidades, para os estados representados na interseção entre o plano (x, y, 0) e a esfera de Bloch.

Insistimos que não se pode calcular exatamente os valores de $|\alpha|$ ou $|\beta|$, em (1.4), mesmo que haja uma grande quantidade de estados $|\psi\rangle$ de mesmo valor. Vejamos por quê. Após serem feitas repetidas medidas dos estados com valores iguais a $|\psi\rangle$, teremos apenas os resultados $|0\rangle$ ou $|1\rangle$. Através da quantidade de $|0\rangle$´s e $|1\rangle$´s encontrados, teremos um valor aproximado para os valores $|\alpha|^2$ e $|\beta|^2$. Não podemos garantir sua exatidão, pois trata-se de probabilidades. E mais, se para sabermos o valor dos "coeficientes" de um simples q-bit, com uma precisão razoável, precisássemos de um número enorme de medidas repetidas de q-bits com mesmo valor, provavelmente haveria pouco interesse em computadores quânticos.

Essa seria uma situação paradoxal, pois apenas medindo estados que forneçam os resultados $|0\rangle$ ou $|1\rangle$, não ultrapassaríamos os marcos da computação clássica. Ou seja, apesar da quantidade infinita de informação que um q-bit guardaria em potencial, apenas dois valores seriam acessados por nós. No entanto, há outro tipo de fenômeno que ocorre com um estado quântico, além daquele ocasionado por sua medida. A mecânica quântica também nos diz que a evolução no tempo de um sistema quântico isolado é descrita matematicamente por uma transformação linear [16]. Ora, sistemas quânticos isolados são descritos por vetores unitários, e, como sabemos da álgebra linear, as funções que transformam vetores unitários em vetores unitários do mesmo espaço vetorial são as *transformações unitárias*.

Transformações lineares unitárias U podem ser definidas (há outras definições equivalentes) como aquelas que atendam à seguinte propriedade:

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

onde $U^\dagger = (U^*)^T$, com $*$ indicando a conjugação complexa, e T indicando a transposição matricial. $U^\dagger$ é denominada *transformação adjunta* de U. Desse ponto em diante, faremos referência indistintamente à transformação U e à matriz que a representa usando a mesma notação, salvo indicação explícita. Usaremos, também, o termo operador com esse mesmo significado. Com isso, quando escrevermos $U|\psi\rangle$, estaremos falando tanto da aplicação de U, quanto da multiplicação da matriz U pelo estado $|\psi\rangle$.

Resumindo: temos, então, duas interações básicas de um computador quântico com os dados de entrada: transformação unitária e medida. A primeira, atuando no nível quântico, e a segunda, fazendo a ligação entre o mundo quântico e o clássico.

1.2.2 Produto tensorial

Para considerarmos estados com mais de um q-bit, precisamos introduzir o conceito de *produto tensorial*. Há vários graus de generalidade para a introdução dessa definição. Usaremos, aqui, a mais simples e que será plenamente suficiente para os nossos propósitos.

O produto tensorial de dois estados

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{bmatrix} \quad \text{e} \quad |\varphi\rangle = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_p \end{bmatrix},$$

denotado por $|\psi\rangle \otimes |\varphi\rangle$, tem como resultado o estado $|\chi\rangle$ com mp-linhas, dado por

$$|\chi\rangle = \begin{bmatrix} \psi_1\varphi_1 \\ \psi_1\varphi_2 \\ \vdots \\ \psi_1\varphi_p \\ \psi_2\varphi_1 \\ \psi_2\varphi_2 \\ \vdots \\ \psi_2\varphi_p \\ \vdots \\ \psi_m\varphi_1 \\ \psi_m\varphi_2 \\ \vdots \\ \psi_m\varphi_p \end{bmatrix}, \tag{1.7}$$

onde $\psi_i\varphi_j$ é o produto usual dos complexos.

Usaremos, também, outras notações mais simplificadas para o produto tensorial $|\psi\rangle \otimes |\varphi\rangle$. São elas: $|\psi\rangle|\varphi\rangle$, $|\psi, \varphi\rangle$ e $|\psi\varphi\rangle$. Por exemplo:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

e

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

Note que o produto tensorial não é comutativo.

O produto tensorial pode ser estendido para matrizes quaisquer. Dadas as matrizes $A \in \mathbb{C}^{m\times n}$ e $B \in \mathbb{C}^{p\times q}$, a matriz $A \otimes B \in \mathbb{C}^{mp\times nq}$ é definida por

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}, \tag{1.8}$$

onde A_{ij} é o elemento da linha i e da coluna j de A. De forma mais precisa, porém mais criptográfica, cada elemento da matriz $A \otimes B$ é definido por

$$(A \otimes B)_{rs} = A_{ij}B_{kl}, \tag{1.9}$$

onde $r = (i-1)p + k$ e $s = (j-1)q + l$, com os índices variando da seguinte forma: $1 \leq i \leq m$, $1 \leq j \leq n$, $1 \leq k \leq p$, $1 \leq l \leq q$.

Por exemplo, se

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

então

$$A \otimes B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

A seguir, damos algumas propriedades do produto tensorial que serão utilizadas ao longo do texto (considere $z \in \mathbb{C}$, v, v_1, $v_2 \in \mathbb{C}^n$ e w, w_1, $w_2 \in \mathbb{C}^m$):

1. $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$,
2. $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = (|v_1\rangle \otimes |w\rangle) + (|v_2\rangle \otimes |w\rangle)$,
3. $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = (|v\rangle \otimes |w_1\rangle) + (|v\rangle \otimes |w_2\rangle)$.

Exercício 1.3 Demonstre as propriedade 1, 2 e 3 do produto tensorial.

Dadas duas transformações lineares A e B, podemos definir um novo operador linear, $A \otimes B$, por

$$(A \otimes B)(|u\rangle \otimes |w\rangle) = A|u\rangle \otimes B|v\rangle, \tag{1.10}$$

desde que garantidas as dimensões corretas para possibilitar as multiplicações das matrizes pelos vetores.

Ainda, introduzindo mais algumas notações, diremos que $|\psi\rangle^{\otimes n}$ e $A^{\otimes n}$ são os produtos tensoriais de $|\psi\rangle$, por ele próprio n vezes, e de A, por ela própria n vezes, respectivamente.

Vejamos, agora, a descrição de um estado genérico $|\psi\rangle$ de 2 q-bits. Esse será uma superposição dos estados $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$ (estamos usando a notação simplificada para o produto tensorial entre dois estados de 1 q-bit), ou seja,

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle, \tag{1.11}$$

onde

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Visando a reduzir a notação, podemos considerar os zeros e uns que aparecem na equação (1.11) como números binários, e assim,

$$|00\rangle,\ |01\rangle,\ |10\rangle,\ |11\rangle$$

podem ser abreviados por

$$|0\rangle,\ |1\rangle,\ |2\rangle,\ |3\rangle,$$

usando a notação decimal. É claro que o $|0\rangle$ acima não é o mesmo que aparece na definição de um q-bit, pois têm dimensões diferentes. Em cada caso, o contexto esclarecerá a que situação estamos nos referindo.

Em geral, um estado $|\psi\rangle$ de n q-bits é uma superposição de 2^n estados da base computacional $\{|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle\}$, dada por

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

com as amplitudes α_i atendendo a

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1.$$

Como havíamos comentado anteriormente, a medição do estado genérico $|\psi\rangle$ produz um resultado $|i_0\rangle$ com probabilidade $|\alpha_{i_0}|^2$, com $0 \leq i_0 \leq 2^n - 1$. Usualmente, a medida é realizada q-bit a q-bit, produzindo zeros e uns que são lidos em conjunto, gerando a saída $|i_0\rangle$. Repetiremos, aqui, uma propriedade central do processo de medida. O estado $|\psi\rangle$, antes da medição, é inacessível, a não ser que ele pertença à base computacional. O procedimento de medida altera inevitavelmente $|\psi\rangle$, forçando-o a um colapso para algum dos vetores da base computacional. Este colapso, como vimos, é não-determinístico, com probabilidades dadas pelos quadrados dos módulos das amplitudes de $|\psi\rangle$.

Consideremos, agora, outro conceito fundamental em computação quântica: o *emaranhamento*. Um estado de 2 q-bits pode ou não ser o resultado do produto tensorial de estados de 1 q-bit. Vejamos. Considere os estados de 1 q-bit

$$|\varphi\rangle = a|0\rangle + b|1\rangle$$

e

$$|\psi\rangle = c|0\rangle + d|1\rangle,$$

onde $a, b, c, d \in \mathbb{C}$. O estado definido pelo produto tensorial de $|\varphi\rangle$ e $|\psi\rangle$ é

$$\begin{aligned} |\varphi\rangle \otimes |\psi\rangle &= (a|0\rangle + b|1\rangle) \otimes (c|0\rangle + d|1\rangle) \\ &= ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle. \end{aligned} \tag{1.12}$$

Observe que um estado de 2 q-bits genérico (1.11) é da forma (1.12) se, e somente se,

$$\begin{aligned} \alpha &= ac, \\ \beta &= ad, \\ \gamma &= bc, \\ \delta &= bd. \end{aligned}$$

Dessas igualdades, temos que

$$\frac{\alpha}{\beta} = \frac{c}{d} \quad \text{e} \quad \frac{\gamma}{\delta} = \frac{c}{d}.$$

Ou seja,

$$\alpha\delta = \beta\gamma.$$

Logo, um estado de 2 q-bits, em geral, não é o produto tensorial de estados de 1 q-bit. Quando isso acontece, dizemos que o estado está emaranhado. Por exemplo, o estado $|01\rangle$ pode, obviamente, ser descrito como produto tensorial dos estados $|0\rangle$ e $|1\rangle$, isto é,

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

No entanto, o estado

$$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

é um estado emaranhado, pois não pode ser descrito como produto tensorial de estados de 1 q-bit.

1.2.3 Produtos interno e externo

Podemos definir o *produto interno* entre os estados $|\varphi\rangle$, $|\psi\rangle \in \mathbb{C}^n$, denotado por $\langle\varphi|\psi\rangle$, como sendo o produto matricial entre $|\varphi\rangle^\dagger$ e $|\psi\rangle$, ou seja,

$$\langle\varphi|\psi\rangle = (|\varphi\rangle)^\dagger|\psi\rangle = \sum_{i=1}^{n} \varphi_i^* \psi_i \,. \tag{1.13}$$

O estado $|\varphi\rangle^\dagger$ é chamado *dual* de $|\varphi\rangle$ e denotado por $\langle\varphi|$ ($|\varphi\rangle$ e $\langle\varphi|$ são denominados *ket* e *bra*, respectivamente).

O produto interno satisfaz às seguintes propriedades:

1. $\langle\psi|\varphi\rangle = \langle\varphi|\psi\rangle^*$,
2. $\langle\varphi|(a|u\rangle + b|v\rangle)\rangle = a\langle\varphi|u\rangle + b\langle\varphi|v\rangle$,
3. $\langle\varphi|\varphi\rangle > 0$ (se $|\varphi\rangle \neq 0$),

com $a, b \in \mathbb{C}$ e $|\varphi\rangle, |\psi\rangle, |u\rangle, |v\rangle \in \mathbb{C}^n$.

EXERCÍCIO 1.4 Demonstre as propriedades 1, 2 e 3 do produto interno.

A *norma* de um estado $|\varphi\rangle$ pode, então, ser definida por

$$|| \, |\varphi\rangle \, || = \sqrt{\langle\varphi|\varphi\rangle}.$$

Podemos, também, definir o *produto externo* entre os estados $|\varphi\rangle \in \mathbb{C}^m$ e $|\psi\rangle \in \mathbb{C}^n$, denotado por $|\varphi\rangle\langle\psi|$, como sendo o produto matricial de $|\varphi\rangle$ por $\langle\psi|$, ou seja,

$$|\varphi\rangle\langle\psi| = |\varphi\rangle(|\psi\rangle)^\dagger.$$

Note que $|\varphi\rangle\langle\psi|$ é uma matriz de ordem $m \times n$.

Como exemplos das definições acima, considere os estados de 1 q-bit

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

e

$$|\psi\rangle = \gamma|0\rangle + \delta|1\rangle.$$

Temos, então,

$$\langle\varphi|\psi\rangle \quad = \quad \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \alpha^*\gamma + \beta^*\delta,$$

para o produto interno, e

$$|\varphi\rangle\langle\psi| \quad = \quad \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \begin{bmatrix} \gamma^* & \delta^* \end{bmatrix} = \begin{bmatrix} \alpha\gamma^* & \alpha\delta^* \\ \beta\gamma^* & \beta\delta^* \end{bmatrix},$$

para o produto externo.

EXERCÍCIO 1.5 Demonstre que, dados dois vetores $|\varphi\rangle, \ |\psi\rangle \in \mathbb{C}^n$, temos

$$(|\psi\rangle\langle\psi|)|\varphi\rangle = \langle\psi|\varphi\rangle|\psi\rangle. \tag{1.14}$$

Usando o produto interno, podemos definir o *ângulo* θ entre dois vetores unitários $|\varphi\rangle$, $|\psi\rangle \in \mathbb{R}^n$ por

$$\theta = \arccos(\langle\varphi|\psi\rangle). \tag{1.15}$$

Observe que, usando essa definição, $\theta \in [0, \pi]$.

Com os conceitos apresentados até aqui, podemos dar uma representação para um computador quântico (Figura 1.6), generalizando o computador clássico, apresentado na Figura 1.1. Os bits de entrada são substituídos por estados de 1 q-bit e a função f é substituída por um operador unitário U que, em geral, é o resultado da composição de vários outros operadores unitários. O resultado da computação é dado pela medida de cada q-bit de saída.

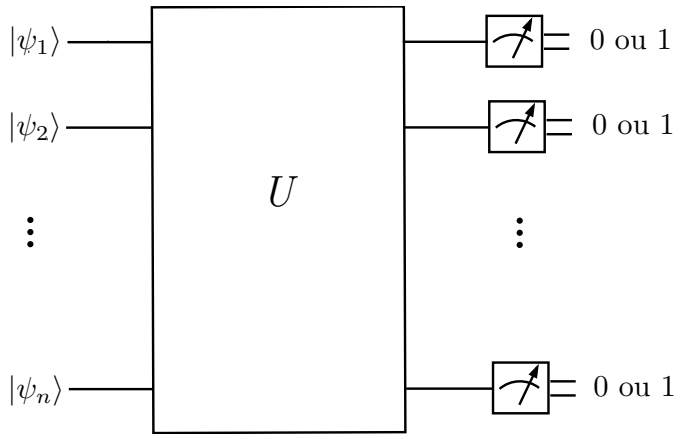

Figura 1.6: Esquema genérico para um computador quântico.

A priori, usando n q-bits, existe a possibilidade de um número infinito de operadores unitários U, representados por matrizes com $2^n \times 2^n$ entradas. Na prática, há que se levar os erros em conta, o que diminui o número de circuitos implementáveis. Mesmo assim, os graus de liberdade são maiores que no computador clássico. Cada operador U é implementado com portas formando circuitos quânticos, assunto do próximo capítulo.

Capítulo 2

Circuitos Quânticos

A representação gráfica de circuitos clássicos é, de certa forma, próxima da realidade física do circuito implementado. Por exemplo, linhas correspondem a fios e bifurcações significam que a corrente elétrica passa por ambos os fios. Nos circuitos quânticos, os fenômenos ocorrem de outra forma, como veremos.

2.1 Notação e Convenções

Para apresentar as convenções usadas em circuitos quânticos, vamos utilizar um circuito (porta U-controlada) em que a entrada e a saída são um estado de 2 q-bits (Figura 2.1).

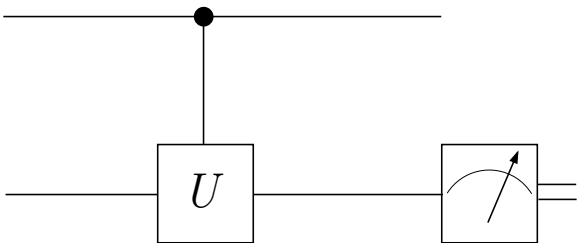

Figura 2.1: Porta quântica U-controlada.

Entrada: pode ser o produto tensorial entre os q-bits de entrada ou um estado emaranhado (os q-bits não devem ser considerados individualmente).

Linhas horizontais: as linhas que aparecem não são necessariamente fios. Elas representam a evolução de um q-bit, podendo ser apenas a passagem do tempo ou, por exemplo, o deslocamento de um fóton.

Sentido: o circuito descreve a evolução do sistema quântico no tempo, da esquerda para a direita. Com isso, não há sentido em aparecer retroalimentação, que pode ocorrer em um circuito clássico.

Linhas verticais: o segmento vertical que aparece unindo os símbolos • e $\boxed{U}$ informa que o circuito atua simultaneamente nos dois q-bits. A linha vertical representa o sincronismo, e não o envio de informação. Portanto, não são permitidas nem junções, nem bifurcações de q-bits.

Controle: o símbolo • indica que o q-bit representado nessa linha é um q-bit de controle, ou seja, caso esteja no estado $|1\rangle$, a porta U realiza a operação; caso esteja no estado $|0\rangle$, a porta U não realiza operação alguma. Caso o q-bit de controle seja um estado superposto ou os 2 q-bits estejam emaranhados, não é possível compreender o comportamento individual do q-bit de controle e do q-bit alvo. Devemos considerar a ação do operador unitário, que representa todo o circuito, atuando simultaneamente nos 2 q-bits.

Saída: os q-bits que compõem a saída do circuito podem ou não ser medidos. Como o q-bit inferior está sendo medido (o símbolo de medida está indicado na Figura 2.1), o resultado será 0 ou 1.

Vistas as principais convenções, vamos apresentar algumas portas quânticas. Comecemos por portas de 1 q-bit. No caso clássico, há apenas uma possibilidade: a porta NOT. O mesmo não ocorre nos circuitos quânticos, como veremos.

Antes de prosseguir, façamos uma observação. A importância do estudo de portas lógicas em computação quântica baseia-se no fato de que toda matriz unitária 2×2 pode ser representada por um circuito quântico de 1 q-bit e vice-versa [16]. Sendo assim, a evolução no tempo de um sistema quântico isolado, dado por um q-bit, pode ser representada tanto matematicamente (por uma transformação unitária) quanto logicamente (por um circuito quântico).

2.2 Porta NOT Quântica

No caso clássico, a porta NOT troca o 1 por 0 e vice-versa. A generalização para o caso quântico é dada por um operador X que satisfaz

$$X|0\rangle = |1\rangle \quad \text{e} \quad X|1\rangle = |0\rangle. \tag{2.1}$$

Com isso, verifica-se facilmente que a representação matricial do operador X é dada por

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

Exercício 2.1 Demonstre que X é um operador unitário.

Com a porta NOT quântica, temos situações sem contrapartida no caso clássico, pois, se a entrada $|\varphi\rangle$ for uma superposição dos estados $|0\rangle$ e $|1\rangle$,

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

a saída será

$$X|\varphi\rangle = \beta|0\rangle + \alpha|1\rangle.$$

A porta X é apenas uma da portas de 1 q-bit, já que há infinitas matrizes unitárias 2×2.

2.3 Porta Hadamard

Uma outra porta de 1 q-bit, largamente utilizada, é a porta Hadamard H, definida pelo operador

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.2}$$

EXERCÍCIO 2.2 Demonstre que H é um operador unitário.

Aplicando H no estado $|0\rangle$, obtemos

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

que é uma superposição dos estados $|0\rangle$ e $|1\rangle$, onde a probabilidade de se obter um dos estados, ao se fazer uma medida do estado $H|0\rangle$, é a mesma: 50%. Aplicando o operador H em cada q-bit de um registrador com 2 q-bits no estado $|00\rangle$, temos:

$$\begin{aligned} H^{\otimes 2}|00\rangle &= H|0\rangle \otimes H|0\rangle \\ &= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \\ &= \frac{1}{2}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right). \end{aligned}$$

Em notação decimal,

$$H^{\otimes 2}|00\rangle = \frac{1}{2}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right).$$

Generalizando para estados com n q-bits, obtemos:

$$\begin{aligned} H^{\otimes n}|0...0\rangle &= (H|0\rangle)^{\otimes n} \\ &= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{N-1} |i\rangle. \end{aligned}$$

Esse resultado será importante no algoritmo de Grover (Capítulo 3).

EXERCÍCIO 2.3 Aplique o operador H em um estado superposto qualquer e interprete o resultado.

2.4 Porta de Fase ou Porta S

A matriz unitária associada à porta S é

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix},$$

onde i é a unidade imaginária ($i^2 = -1$). A porta S pode também ser representada por

$$S = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/2) \end{bmatrix},$$

já que $\exp(i\pi/2) = \cos(\pi/2) + i\ \mathrm{sen}(\pi/2) = i$.

Aplicando S em um estado genérico

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

obtemos

$$S|\psi\rangle = \alpha|0\rangle + i\beta|1\rangle.$$

Note que, se for feita uma medida do estado $S|\psi\rangle$, as probabilidades de se obter os estados $|0\rangle$ ou $|1\rangle$ serão as mesmas, comparadas com uma medida realizada sobre o estado $|\psi\rangle$. Isso não acontece, por exemplo, usando a porta H.

2.5 Porta $\pi/8$ ou Porta T

A matriz unitária associada à porta T é

$$T = \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{bmatrix},$$

que poderia ser representada, também, na forma

$$T = \exp(i\pi/8) \begin{bmatrix} \exp(-i\pi/8) & 0 \\ 0 & \exp(i\pi/8) \end{bmatrix}.$$

Aplicando T em um estado genérico

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

obtemos

$$T|\psi\rangle = \alpha|0\rangle + \exp(i\pi/4)\beta|1\rangle.$$

Também, nesse caso, se for feita uma medida do estado $T|\psi\rangle$, as probabilidades de se obter os estados $|0\rangle$ ou $|1\rangle$ serão as mesmas, comparadas com uma medida realizada sobre o estado $|\psi\rangle$.

2.6 Porta CNOT Quântica

Outra porta, essa atuando em estados de 2 q-bits, é a contrapartida quântica do circuito clássico apresentado anteriormente na Figura 1.3. Ela tem 2 q-bits de entrada, o de controle e o alvo (Figura 2.2). Uma porta controlada, como já vimos (Figura 2.1), age dependendo do valor do q-bit de controle. Ela é "ativada" se o q-bit de controle estiver no estado $|1\rangle$, e nada faz, se ele estiver no estado $|0\rangle$. Essa descrição é adequada apenas quando o q-bit de controle está nos estados $|0\rangle$ ou $|1\rangle$. Entretanto, o que distingue a porta CNOT quântica da clássica é que, na porta CNOT quântica, os q-bits alvo e de controle podem ser estados superpostos, e, além disso, os dois q-bits podem estar emaranhados.

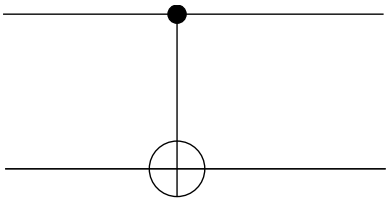

Figura 2.2: Porta CNOT quântica.

A ação da porta CNOT pode ser caracterizada pelas transformações operadas nos elementos da base computacional associada, ou seja,

$$\begin{aligned} |00\rangle &\rightarrow |00\rangle, \\ |01\rangle &\rightarrow |01\rangle, \\ |10\rangle &\rightarrow |11\rangle, \\ |11\rangle &\rightarrow |10\rangle. \end{aligned} \tag{2.3}$$

Note que podemos representar essa ação na base computacional de forma mais esquemática por

$$|i, j\rangle \rightarrow |i, i \oplus j\rangle, \tag{2.4}$$

onde i, $j \in \{0, 1\}$ e $\oplus$ é a adição módulo 2.

Para obtermos a matriz U_{CNOT} associada à porta CNOT, basta usarmos os valores dados em (2.3), isto é,

$$U_{CNOT}\begin{bmatrix}1\\0\\0\\0\end{bmatrix}=\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \qquad U_{CNOT}\begin{bmatrix}0\\1\\0\\0\end{bmatrix}=\begin{bmatrix}0\\1\\0\\0\end{bmatrix},$$

$$U_{CNOT}\begin{bmatrix}0\\0\\1\\0\end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}, \qquad U_{CNOT}\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\0\\1\\0\end{bmatrix},$$

que resulta em

$$U_{\text{CNOT}}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}. \tag{2.5}$$

Exercício 2.4 Demonstre que U_{CNOT} é um operador unitário.

Exercício 2.5 Dê um exemplo de estado emaranhado produzido pela porta CNOT.

Um resultado importante sobre circuitos quânticos é que qualquer operador unitário pode ser representado usando portas CNOT e portas de 1 q-bit [16].

Exercício 2.6 Demonstre que a matriz U_{CNOT} não pode ser descrita como produto tensorial de matrizes 2×2.

Exercício 2.7 Demonstre que a porta CNOT não pode ser descrita usando portas de 1 q-bit.

2.7 Porta Toffoli Quântica

A próxima porta a ser considerada é a correspondente quântica da porta Toffoli (Figura 1.4). Também é uma porta controlada, só que nesse caso, com dois q-bits de controle (Figura 2.3). Sua ação na base computacional associada pode ser representada por

$$|i,j,k\rangle \rightarrow |i,j,k \oplus ij\rangle,$$

onde i, j, $k \in \{0,1\}$ e $\oplus$ é a adição módulo 2. Observe que, nesse caso, a base computacional possui 8 elementos.

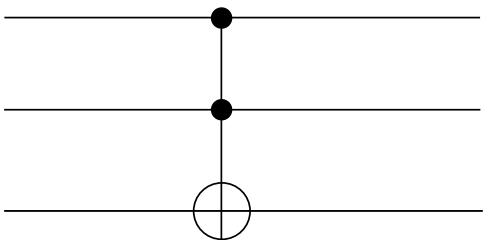

Figura 2.3: Porta Toffoli quântica.

EXERCÍCIO 2.8 Exiba a matriz associada à porta Toffoli quântica.

EXERCÍCIO 2.9 Analise se a matriz associada à porta Toffoli quântica pode ser representada pelo produto tensorial de matrizes quadradas de dimensões diferentes de 1×1.

A porta Toffoli é usada para simplificar a representação de circuitos quânticos. Como já sabemos, ela pode ser descrita usando portas de 1 q-bit e portas CNOT. Uma representação possível é dada na Figura 2.4 (há outras representações [19]).

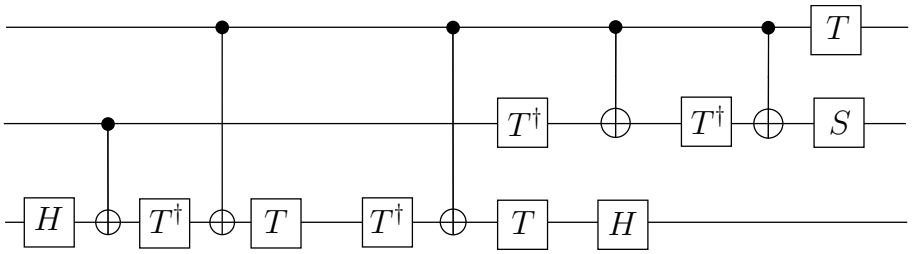

Figura 2.4: Decomposição da porta Toffoli em portas de 1 q-bit e portas CNOT.

EXERCÍCIO 2.10 Faça a evolução dos estados da base computacional, na representação da porta Toffoli da Figura 2.4.

Para simplificar ainda mais a representação de circuitos quânticos, temos também a porta Toffoli generalizada (Figura 2.5), que será utilizada nos capítulos seguintes.

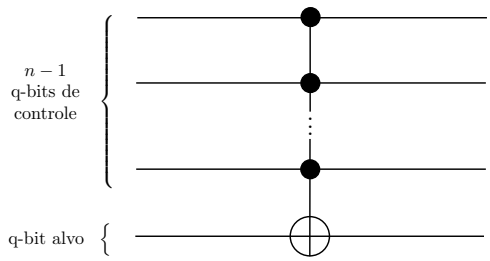

Figura 2.5: Porta Toffoli generalizada.

A decomposição da porta Toffoli generalizada, em termos de portas Toffoli simples, é mostrada na Figura 2.6. Os $n-2$ q-bits de trabalho são q-bits extras, cujas entradas são conhecidas antecipadamente. São utilizados para simplificar a decomposição.

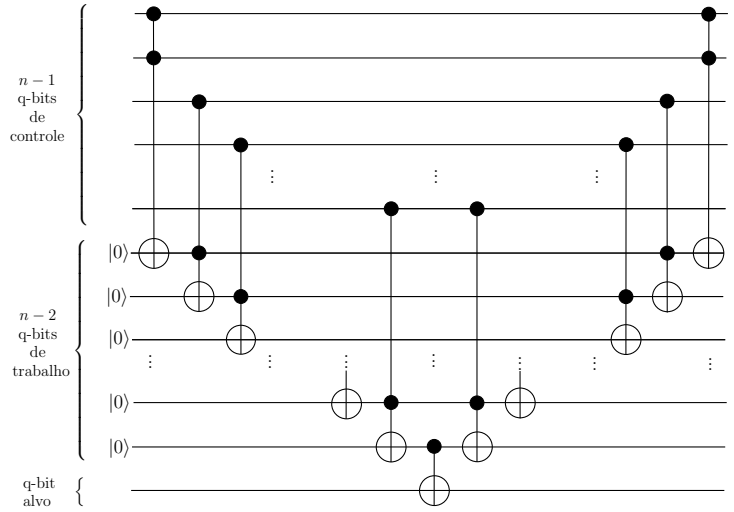

Figura 2.6: Porta Toffoli generalizada decomposta em portas Toffoli simples.

Exercício 2.11 Analise as saídas da porta Toffoli generalizada (Figura 2.5) e as saídas da sua decomposição (Figura 2.6), considerando, na entrada, elementos da base computacional.

Capítulo 3

Algoritmo de Grover

3.1 Introdução

Considere o seguinte problema: temos uma lista não ordenada com N elementos e desejamos encontrar um elemento específico que está na lista. Classicamente, deveríamos testar elemento a elemento. No pior caso possível, precisaríamos realizar N testes. Como veremos, usando as propriedades da mecânica quântica, a quantidade de "testes" necessários para a identificação do elemento procurado será proporcional a $\sqrt{N}$. Este resultado será obtido usando o algoritmo de Grover [12, 13].

Para representar matematicamente o problema, vamos supor que a busca será realizada sobre a lista $\{0, 1, ..., N-1\}$, onde $N = 2^n$ para algum número natural n, e que a função

$$f : \{0, 1, ..., N-1\} \to \{0, 1\},$$

definida por

$$f(i) = \begin{cases} 1, \text{ se } i = i_0, \\ 0, \text{ se } i \neq i_0, \end{cases} \tag{3.1}$$

será utilizada para o reconhecimento do elemento procurado i_0 (assumiremos que existe um único elemento $i \in \{0, 1, ..., N-1\}$ tal que $f(i) = 1$). Dessa forma, o custo computacional para resolver o problema está associado ao número de vezes que a função f deve ser "utilizada". Imagine a função f como sendo um oráculo que está à disposição para informar se um dado elemento é ou não o elemento procurado.

O algoritmo de Grover utiliza dois registradores quânticos (Figura 3.1): o primeiro, com n q-bits, inicializado no estado $|0...0\rangle$, e o segundo, com 1 q-bit, inicializado no estado $|1\rangle$. O primeiro registrador está relacionado aos elementos da lista onde será feita a busca, enquanto que o segundo é um registrador que terá um papel fundamental, como veremos. A cada elemento i da lista $\{0, 1, ..., N-1\}$, associaremos o estado $|i\rangle$ de n q-bits.

3.2 Operadores do Algoritmo

Antes da execução propriamente dita do algoritmo, o primeiro registrador é alterado para formar uma superposição de todos os estados associados aos elementos da lista. Isso pode ser obtido aplicando o operador Hadamard H (2.2) em cada q-bit do primeiro registrador (Figura 3.1).

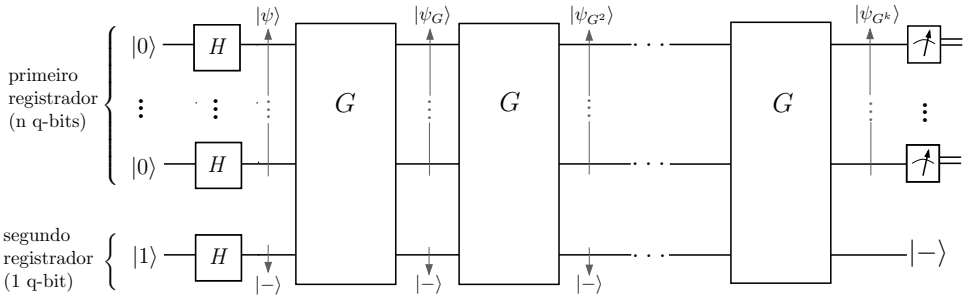

Figura 3.1: Esquema genérico para o algoritmo de Grover (G é um operador unitário que será definido mais adiante).

A superposição obtida será denotada por $|\psi\rangle$, ou seja,

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle. \tag{3.2}$$

Observe que aplicando n vezes o operador H, obtemos uma superposição de $N = 2^n$ estados com mesma amplitude.

Para completar a inicialização do algoritmo, o operador H também é aplicado sobre o estado inicial do segundo registrador (Figura 3.1). Denotando o resultado por $|-\rangle$, temos:

$$|-\rangle = H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{3.3}$$

Já sabemos que qualquer alteração de um sistema quântico isolado (que não seja uma medida) é descrita por um operador unitário. Para "representar" quanticamente a função f, em (3.1), utilizada para a identificação do elemento procurado, imaginemos, então, um operador linear U_f que transforme $|i\rangle$ em $|f(i)\rangle$, onde $|i\rangle$ é o estado de n q-bits do primeiro registrador. Como U_f deve ser unitário, a "entrada" e a "saída" de U_f devem ter a mesma dimensão. Considere, então,

$$|i\rangle|0\rangle \xrightarrow{U_f} |i\rangle|f(i)\rangle, \tag{3.4}$$

onde, na "entrada" e na "saída", o primeiro registrador tem n q-bits e o segundo apenas 1 q-bit. Usando (3.4), temos:

$$U_f\left(|i\rangle|0\rangle\right) = \begin{cases} |i\rangle|1\rangle, \text{ se } i = i_0, \\ |i\rangle|0\rangle, \text{ se } i \neq i_0. \end{cases} \tag{3.5}$$

Ou seja, o operador U_f altera o estado do segundo registrador quando o primeiro registrador representa o elemento procurado. Para completar a definição, precisamos definir o valor de $U_f\left(|i\rangle|1\rangle\right)$. Mantendo a mesma idéia, definimos:

$$U_f\left(|i\rangle|1\rangle\right) = \begin{cases} |i\rangle|0\rangle, \text{ se } i = i_0, \\ |i\rangle|1\rangle, \text{ se } i \neq i_0. \end{cases} \tag{3.6}$$

Com isso, U_f fica bem definido, pois, sendo um operador linear, basta defini-lo nos elementos da base. Note que a base é formada usando o produto tensorial. Por exemplo, para uma lista com 4 elementos (o primeiro registrador terá 2 q-bits), a base será

$$\{|0\rangle|0\rangle, |0\rangle|1\rangle, |1\rangle|0\rangle, |1\rangle|1\rangle, |2\rangle|0\rangle, |2\rangle|1\rangle, |3\rangle|0\rangle, |3\rangle|1\rangle\},$$

ou melhor,

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}.$$

Exercício 3.1 Exiba a matriz que representa U_f.

Para facilitar os cálculos a seguir, representaremos (3.5) e (3.6) de uma única maneira, isto é,

$$U_f\left(|i\rangle|j\rangle\right) = |i\rangle|j \oplus f(i)\rangle, \tag{3.7}$$

onde $|i\rangle$ é o estado de n q-bits do primeiro registrador ($i \in \{0, 1, \ldots, N-1\}$), $|j\rangle$ é o estado de 1 q-bit do segundo registrador ($j \in \{0, 1\}$) e $\oplus$ é a soma módulo 2. Note que $U_f \in \mathbb{C}^{(2^{n+1} \times 2^{n+1})}$.

Exercício 3.2 Demonstre que U_f é um operador unitário.

O operador U_f foi definido para simular quanticamente o papel da função f (3.1). Para identificar o elemento procurado i_0, bastaria aplicar U_f em cada estado associado aos elementos da lista e manter o segundo registrador no estado $|0\rangle$ ou $|1\rangle$. Quando o estado do segundo registrador fosse alterado, saberíamos que o elemento buscado teria sido encontrado. Neste caso, o estado do primeiro registrador seria $|i_0\rangle$. No entanto, isso não proporcionaria ganho algum em relação ao caso clássico, usando a função f. O que vai fazer diferença é que podemos também aplicar U_f em estados superpostos. Vejamos.

O próximo passo do algoritmo é aplicar o operador U_f sobre o estado $|\psi\rangle|-\rangle$, resultante da inicialização (Figura 3.2). Ou seja,

$$U_f\left(|\psi\rangle|-\rangle\right) = U_f\left(\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\right)|-\rangle\right).$$

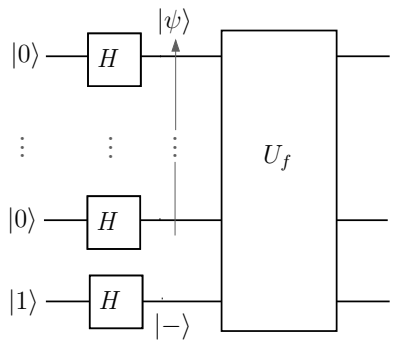

Figura 3.2: Aplicação do operador U_f sobre o estado $|\psi\rangle|-\rangle$.

Usando a distributividade do produto tensorial em relação à adição de vetores,

$$U_f\left(|\psi\rangle|-\rangle\right) = U_f\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|-\rangle\right).$$

Da linearidade do operador U_f,

$$U_f\left(|\psi\rangle|-\rangle\right) = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}U_f\left(|i\rangle|-\rangle\right).$$

Substituindo a definição do estado $|-\rangle$, dada em (3.3), na expressão acima, obtemos:

$$\begin{aligned} U_f\left(|\psi\rangle|-\rangle\right) &= \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}U_f\left(|i\rangle\left(\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)\right)\right) \\ &= \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}U_f\left(\frac{1}{\sqrt{2}}\left(|i\rangle|0\rangle-|i\rangle|1\rangle\right)\right). \end{aligned}$$

Novamente, da linearidade de U_f,

$$U_f\left(|\psi\rangle|-\rangle\right) = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\frac{1}{\sqrt{2}}\left(U_f\left(|i\rangle|0\rangle\right)-U_f(|i\rangle|1\rangle)\right).$$

Da definição de U_f, em (3.7), temos:

$$\begin{aligned} U_f\left(|\psi\rangle|-\rangle\right) &= \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\frac{1}{\sqrt{2}}\left(|i\rangle|f(i)\rangle-|i\rangle|1\oplus f(i)\rangle\right) \\ &= \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}\frac{1}{\sqrt{2}}\left(|i\rangle\left(|f(i)\rangle-|1\oplus f(i)\rangle\right)\right). \end{aligned} \tag{3.8}$$

Da definição de f, em (3.1),

$$|i\rangle\left(|f(i)\rangle - |1 \oplus f(i)\rangle\right) = \begin{cases} |i\rangle\left(|1\rangle - |0\rangle\right), \text{ se } i = i_0, \\ |i\rangle\left(|0\rangle - |1\rangle\right), \text{ se } i \neq i_0. \end{cases} \tag{3.9}$$

Substituindo a expressão anterior em (3.8), temos:

$$U_f\left(|\psi\rangle|-\rangle\right) = \frac{1}{\sqrt{N}}\left(\sum_{i=0, i\neq i_0}^{N-1}\left(\frac{1}{\sqrt{2}}\left(|i\rangle\left(|0\rangle - |1\rangle\right)\right)\right) + \frac{1}{\sqrt{2}}\left(|i_0\rangle\left(|1\rangle - |0\rangle\right)\right)\right).$$

Novamente, da definição de $|-\rangle$,

$$\begin{aligned} U_f\left(|\psi\rangle|-\rangle\right) &= \frac{1}{\sqrt{N}}\left(\left(\sum_{i=0, i\neq i_0}^{N-1}|i\rangle|-\rangle\right) - |i_0\rangle|-\rangle\right) \\ &= \frac{1}{\sqrt{N}}\left(\sum_{i=0}^{N-1}(-1)^{f(i)}|i\rangle|-\rangle\right). \end{aligned}$$

Ou ainda,

$$U_f\left(|\psi\rangle|-\rangle\right) = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}(-1)^{f(i)}|i\rangle\right)|-\rangle. \tag{3.10}$$

Note que o estado do segundo registrador não se altera (como visto acima, isso não quer dizer que ele seja desnecessário!). O estado do primeiro registrador continua sendo uma superposição de todos os estados associados aos elementos da lista. Entretanto, a amplitude do elemento procurado foi alterada de $\frac{1}{\sqrt{N}}$ para $-\frac{1}{\sqrt{N}}$.

Após a aplicação do operador U_f, um fato interessante ocorreu. Além da função f ter sido "avaliada" em todos os elementos da lista onde está sendo feita a busca, com apenas uma aplicação de U_f (este fenômeno é conhecido como *paralelismo quântico* [16]), o estado associado ao elemento procurado foi "identificado" como sendo o único que teve sua amplitude alterada. No entanto, essa informação só está disponível quanticamente. Não adiantaria fazer uma medida do primeiro registrador, pois a probabilidade de se obter o elemento procurado é

$$\left|\frac{-1}{\sqrt{N}}\right|^2 = \frac{1}{N}.$$

Antes de prosseguirmos, consideremos a seguinte questão: a aplicação do operador U_f sobre um estado qualquer, no primeiro registrador, ainda mantém o segundo registrador no estado $|-\rangle$? Vejamos.

Seja $|i\rangle$, um estado qualquer da base computacional $\{|0\rangle, |1\rangle, ..., |N-1\rangle\}$. Usando as definições do operador U_f e do estado $|-\rangle$, temos:

$$\begin{aligned} U_f\left(|i\rangle|-\rangle\right) &= U_f\left(|i\rangle\left(\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)\right) \\ &= U_f\left(\frac{1}{\sqrt{2}}\left(|i\rangle|0\rangle - |i\rangle|1\rangle\right)\right) \\ &= \frac{1}{\sqrt{2}}\left(U_f\left(|i\rangle|0\rangle\right) - U_f(|i\rangle|1\rangle)\right) \\ &= \frac{1}{\sqrt{2}}\left(|i\rangle|f(i)\rangle - |i\rangle|1 \oplus f(i)\rangle\right). \end{aligned}$$

Da mesma forma que fizemos no cálculo de $U_f\left(|\psi\rangle|-\rangle\right)$, obtemos:

$$U_f\left(|i\rangle|-\rangle\right) = (-1)^{f(i)}|i\rangle|-\rangle.$$

Ou seja,

$$U_f\left(|i\rangle|-\rangle\right) = \begin{cases} -|i\rangle|-\rangle, \text{ se } i = i_0, \\ |i\rangle|-\rangle, \text{ se } i \neq i_0. \end{cases} \tag{3.11}$$

Usando este resultado e aplicando U_f sobre um vetor unitário qualquer

$$|v\rangle = \sum_{i=0, i\neq i_0}^{N-1} \alpha_i|i\rangle + \alpha_{i_0}|i_0\rangle,$$

gerado pelos elementos da base computacional $\{|0\rangle, |1\rangle, ..., |N-1\rangle\}$, no primeiro registrador, e mantendo o estado $|-\rangle$, no segundo registrador, temos:

$$\begin{aligned} U_f\left(\left(\sum_{i=0, i\neq i_0}^{N-1} \alpha_i|i\rangle + \alpha_{i_0}|i_0\rangle\right)|-\rangle\right) &= U_f\left(\sum_{i=0, i\neq i_0}^{N-1} \alpha_i|i\rangle|-\rangle + \alpha_{i_0}|i_0\rangle|-\rangle\right) \\ &= \sum_{i=0, i\neq i_0}^{N-1} \alpha_i U_f\left(|i\rangle|-\rangle\right) + \alpha_{i_0}U_f\left(|i_0\rangle|-\rangle\right) \\ &= \sum_{i=0, i\neq i_0}^{N-1} \alpha_i|i\rangle|-\rangle - \alpha_{i_0}|i_0\rangle|-\rangle \\ &= \left(\sum_{i=0, i\neq i_0}^{N-1} \alpha_i|i\rangle - \alpha_{i_0}|i_0\rangle\right)|-\rangle. \end{aligned} \tag{3.12}$$

Conclusão: a aplicação de U_f sobre o estado $|v\rangle|-\rangle$ não altera o estado do segundo registrador. Portanto, para simplificar os cálculos, sempre que o estado do segundo registrador for $|-\rangle$, como é o caso no algoritmo de Grover, omitiremos o segundo registrador. É importante destacar que o estado $|-\rangle$ é fundamental no processo de marcação do elemento procurado.

EXERCÍCIO 3.3 Verifique o que acontece se, ao aplicarmos o operador U_f, o estado do segundo registrador não for o estado $|-\rangle$.

Voltemos ao algoritmo. Com o elemento a ser buscado já identificado quanticamente, o próximo passo será aumentar a probabilidade de esse elemento ser obtido, após uma medida.

O novo estado do primeiro registrador será denotado por $|\psi_1\rangle$, isto é,

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle. \tag{3.13}$$

Olhando com mais cuidado o resultado da aplicação de U_f sobre o estado $|v\rangle|-\rangle$, em (3.12), podemos obter uma interpretação geométrica do efeito do operador U_f sobre o primeiro registrador: a aplicação de U_f sobre um vetor unitário qualquer gerado pelos elementos da base computacional $\{|0\rangle, |1\rangle, ..., |N-1\rangle\}$ resulta numa reflexão desse vetor em relação ao subespaço ortogonal a $|i_0\rangle$, gerado por todos os outros elementos da base computacional. Para "visualizar" esse resultado, podemos considerar essa reflexão como uma reflexão em relação à projeção de $|v\rangle$ sobre o subespaço ortogonal a $|i_0\rangle$. Denotando essa projeção pelo vetor unitário $|u\rangle$, temos:

$$|u\rangle = \frac{1}{\sqrt{N-1}} \sum_{i=0, i\neq i_0}^{N-1} |i\rangle. \tag{3.14}$$

EXERCÍCIO 3.4 Demonstre que a projeção de $|\psi\rangle$, definido em (3.2), sobre o subespaço ortogonal a $|i_0\rangle$ pode ser representada por

$$|u\rangle = \frac{\sqrt{N}}{\sqrt{N-1}} |\psi\rangle - \frac{1}{\sqrt{N-1}} |i_0\rangle. \tag{3.15}$$

Para completar a visualização, calculemos os ângulos entre $|\psi\rangle$ e $|i_0\rangle$ e entre $|u\rangle$ e $|i_0\rangle$. Usando o produto interno, temos:

$$\langle\psi|i_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} \langle i|i_0\rangle = \frac{1}{\sqrt{N}} \langle i_0|i_0\rangle = \frac{1}{\sqrt{N}} \tag{3.16}$$

e

$$\langle u|i_0\rangle = \frac{1}{\sqrt{N-1}} \sum_{i=0, i\neq i_0}^{N-1} \langle i|i_0\rangle = 0. \tag{3.17}$$

Ou seja, o ângulo entre $|\psi\rangle$ e $|i_0\rangle$ é menor do que $\pi/2$ rad (se N é grande, o ângulo é quase $\pi/2$ rad) e o ângulo entre $|u\rangle$ e $|i_0\rangle$ é exatamente $\pi/2$ rad. Usando os resultados (3.16), (3.17) e a expressão dada em (3.15), podemos, finalmente, obter uma representação geométrica para a ação do operador U_f sobre o estado $|\psi\rangle$, dada na Figura 3.3.

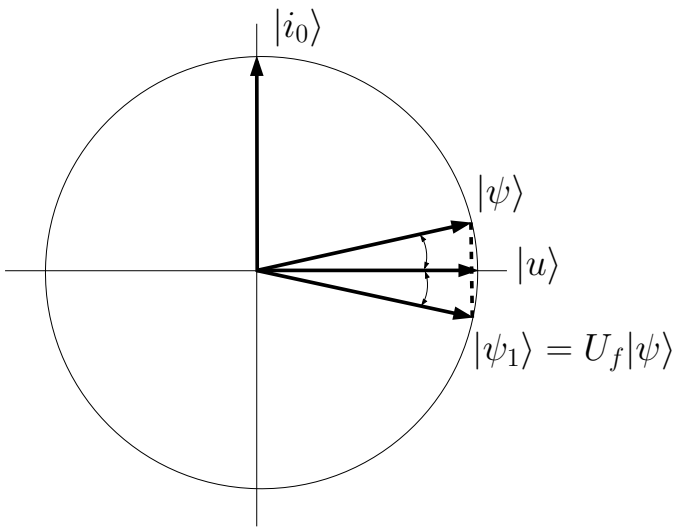

Figura 3.3: Ação de U_f sobre o estado $|\psi\rangle$.

Induzidos por essa representação, poderíamos, então, refletir o vetor $|\psi_1\rangle$ em relação ao vetor $|\psi\rangle$, para aumentar a amplitude do elemento procurado $|i_0\rangle$, em relação à sua amplitude no estado $|\psi\rangle$ (Figura 3.4).

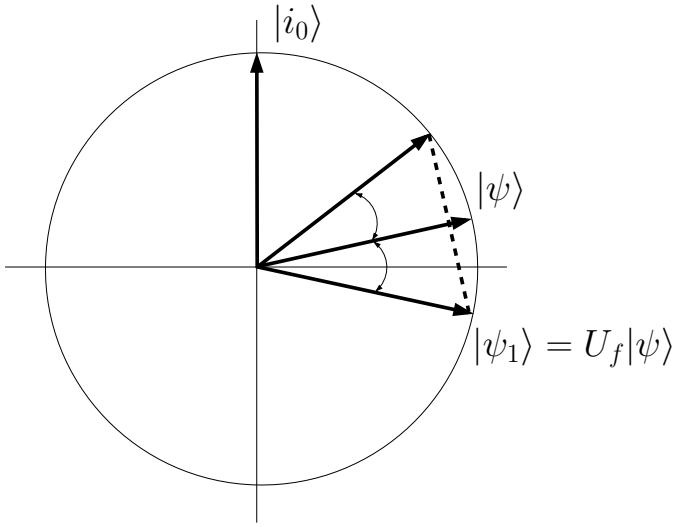

Figura 3.4: Reflexão de $|\psi_1\rangle$ em relação a $|\psi\rangle$.

Uma observação importante: como todas as amplitudes dos estados envolvidos no algoritmo de Grover são números reais, o produto interno sempre resultará em

um número real. Isso possibilita a comparação entre ângulos de dois pares de estados quaisquer. A partir de agora, teremos em mente esse fato.

A projeção de $|\psi_1\rangle$ sobre $|\psi\rangle$ é dada por $\langle\psi|\psi_1\rangle|\psi\rangle$. Motivados pelo losango abaixo (Figura 3.5), vemos que o vetor resultante da reflexão de $|\psi_1\rangle$ em relação a $|\psi\rangle$ pode ser descrito como

$$(2\langle\psi|\psi_1\rangle)\,|\psi\rangle - |\psi_1\rangle. \tag{3.18}$$

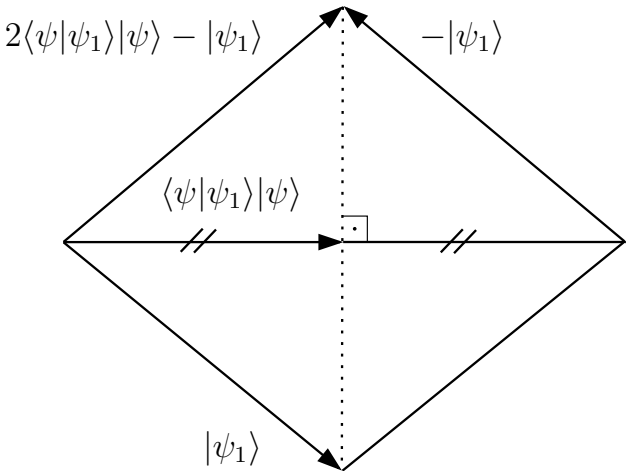

Figura 3.5: Reflexão de $|\psi_1\rangle$ em relação a $|\psi\rangle$.

O que desejamos é obter um novo operador que produza essa reflexão. Usando a propriedade (1.14), podemos reescrever a expressão acima, obtendo:

$$(2\langle\psi|\psi_1\rangle)\,|\psi\rangle - |\psi_1\rangle = (2|\psi\rangle\langle\psi|)\,|\psi_1\rangle - |\psi_1\rangle = (2|\psi\rangle\langle\psi| - I)\,|\psi_1\rangle.$$

Ou seja, o operador que procuramos é

$$2|\psi\rangle\langle\psi| - I, \tag{3.19}$$

onde I é o operador identidade.

O estado resultante do primeiro registrador, após a aplicação do operador U_f, em (3.13), pode ser reescrito como

$$|\psi_1\rangle = |\psi\rangle - \frac{2}{\sqrt{N}}|i_0\rangle, \tag{3.20}$$

onde

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle \tag{3.21}$$

e i_0 é o elemento procurado. Denotando por $|\psi_G\rangle$ (Figura 3.6), o estado resultante da aplicação do operador $2|\psi\rangle\langle\psi| - I$ sobre $|\psi_1\rangle$, e usando (3.20), obtemos:

$$\begin{aligned} |\psi_G\rangle &= (2|\psi\rangle\langle\psi| - I)\,|\psi_1\rangle \\ &= (2|\psi\rangle\langle\psi| - I)\left(|\psi\rangle - \frac{2}{\sqrt{N}}|i_0\rangle\right) \\ &= (2\langle\psi|\psi\rangle)\,|\psi\rangle - \left(\frac{4}{\sqrt{N}}\langle\psi|i_0\rangle\right)|\psi\rangle - |\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle. \end{aligned} \tag{3.22}$$

Substituindo (3.16) em (3.22), temos:

$$|\psi_G\rangle = \frac{N-4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle. \tag{3.23}$$

Esse é, então, o estado do primeiro registrador após a aplicação dos operadores U_f e $2|\psi\rangle\langle\psi| - I$ (o estado do segundo registrador permanece inalterado). A composição desses dois operadores é chamada de *operador de Grover* G, isto é,

$$G = ((2|\psi\rangle\langle\psi| - I) \otimes I)\,U_f. \tag{3.24}$$

O segundo operador identidade aparece, porque o operador $2|\psi\rangle\langle\psi| - I$ é aplicado apenas no primeiro registrador (Figura 3.6).

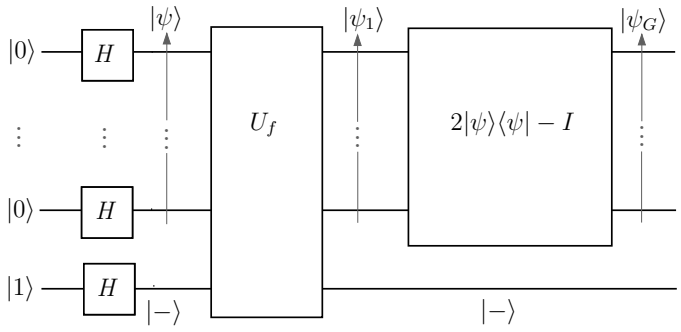

Figura 3.6: Uma aplicação do operador de Grover (G).

EXERCÍCIO 3.5 Demonstre que G é um operador unitário.

De (3.23), obtemos a amplitude do estado $|i_0\rangle$, após a primeira aplicação do operador G:

$$\left(\frac{N-4}{N}\right)\left(\frac{1}{\sqrt{N}}\right) + \frac{2}{\sqrt{N}} = \left(\frac{3N-4}{N\sqrt{N}}\right).$$

Por exemplo, para $N = 4$, a probabilidade de se obter o elemento procurado, após uma medida do estado $|\psi\rangle$, em (3.21), é 25%. Já a probabilidade de se obter o elemento procurado, após uma medida do estado $|\psi_G\rangle$, em (3.23), é 100%. No

entanto, para valores grandes de N, essa probabilidade ainda é pequena. Até agora, o que podemos garantir é que, com uma aplicação do operador G, a amplitude do estado $|i_0\rangle$ é aumentada, em relação à sua amplitude no estado $|\psi\rangle$. E se aplicarmos novamente o operador G sobre o estado $|\psi_G\rangle|-\rangle$? A interpretação geométrica dos operadores U_f e $2|\psi\rangle\langle\psi| - I$ nos induz justamente a isso (Figuras 3.3 e 3.4).

3.3 Custo Computacional do Algoritmo

Como demonstraremos nesta seção, o estado resultante do primeiro registrador, após cada aplicação do operador G, vai se aproximando do estado $|i_0\rangle$. Então, para determinar o custo computacional do algoritmo de Grover, temos que calcular quantas aplicações de G serão necessárias.

Inicialmente, demonstraremos que a aplicação de G^k ($k \in \mathbb{N}$) produz um rotação de $|\psi\rangle$ em direção a $|i_0\rangle$, de $k\theta$ rad, no subespaço gerado pelos vetores $|\psi\rangle$ e $|i_0\rangle$, onde θ é o ângulo entre $|\psi\rangle$ e $G|\psi\rangle$ (Figura 3.7). Para facilitar a leitura, dividiremos a demostração em 4 proposições. A Proposição 1 diz que $G^k|\psi\rangle$ pertence ao subespaço gerado por $|\psi\rangle$ e $|i_0\rangle$, para todo $k \in \mathbb{N}$. A Proposição 2 estabelece que o ângulo entre $G^k|\psi\rangle$ e $G^{k+1}|\psi\rangle$ também é θ, para todo $k \in \mathbb{N}$. Na Proposição 3, demonstramos que G rotaciona $|\psi\rangle$ em direção a $|i_0\rangle$. Finalmente, na Proposição 4, provamos que o sentido da rotação produzida quando G é aplicado sobre $G^k|\psi\rangle$, para todo $k \in \mathbb{N}$, é o mesmo obtido quando G é aplicado sobre $|\psi\rangle$. O subespaço gerado por $|\psi\rangle$ e $|i_0\rangle$ será denotado por Ω e o estado do primeiro registrador de $G^k|\psi\rangle$ será denotado por $|\psi_{G^k}\rangle$. O estado do segundo registrador ($|-\rangle$) será omitido, pois ele é constante durante todo o processo.

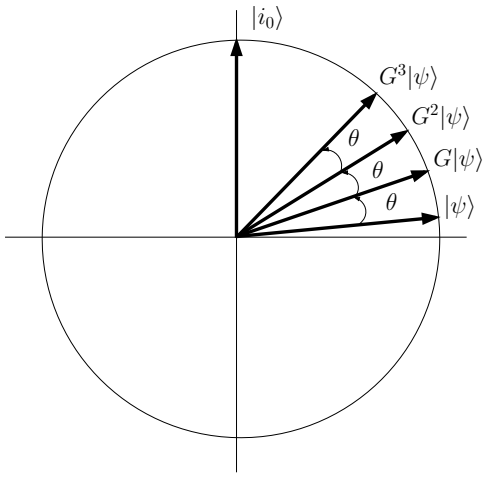

Figura 3.7: Efeito da aplicação do operador G.

Proposição 1 *$G^n|\psi\rangle \in \Omega$, para todo $n \in \mathbb{N}$.*

Prova. A demonstração é por indução. De (3.23), sabemos que

$$G|\psi\rangle = \frac{N-4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle. \tag{3.25}$$

Com isso, temos o resultado para $n = 1$. Suponhamos que, para um dado $k \in \mathbb{N}$,

$$G^k|\psi\rangle \in \Omega.$$

Isto é, existem $\alpha, \beta \in \mathbb{R}$ tais que

$$G^k|\psi\rangle = \alpha|\psi\rangle + \beta|i_0\rangle. \tag{3.26}$$

Temos que provar que

$$G^{k+1}|\psi\rangle \in \Omega.$$

Aplicando o operador G nos dois lados de (3.26), obtemos:

$$G^{k+1}|\psi\rangle = \alpha G|\psi\rangle + \beta G|i_0\rangle. \tag{3.27}$$

Já sabemos que $G|\psi\rangle \in \Omega$. Calculemos $G|i_0\rangle$. Da definição de G, em (3.24), temos:

$$G|i_0\rangle = (2|\psi\rangle\langle\psi| - I)\, U_f|i_0\rangle. \tag{3.28}$$

De (3.11),

$$U_f|i_0\rangle = -|i_0\rangle. \tag{3.29}$$

Substituindo (3.29) em (3.28) e usando (3.16), obtemos:

$$\begin{aligned} G|i_0\rangle &= (2|\psi\rangle\langle\psi| - I)\,(-|i_0\rangle) \\ &= -2\langle\psi|i_0\rangle|\psi\rangle + |i_0\rangle \\ &= -\frac{2}{\sqrt{N}}|\psi\rangle + |i_0\rangle. \end{aligned} \tag{3.30}$$

Ou seja, $G|i_0\rangle \in \Omega$. Como os estados $G|\psi\rangle$ e $G|i_0\rangle$ pertencem a Ω, de (3.27), concluímos que

$$G^{k+1}|\psi\rangle \in \Omega,$$

que finaliza a indução. ■

PROPOSIÇÃO 2 *O ângulo entre $G^k|\psi\rangle$ e $G^{k+1}|\psi\rangle$ é θ rad, para todo $k \in \mathbb{N}$.*

Prova. Usando a definição de ângulo entre dois vetores, dada no Capítulo 1, p. 12, o enunciado deste lema torna-se equivalente a

$$\langle\psi_{G^k}|\psi_{G^{k+1}}\rangle = \cos\theta,\ \forall k \in \mathbb{N}.$$

Reescrevendo, temos

$$\begin{aligned} \langle\psi_{G^k}|\psi_{G^{k+1}}\rangle &= \langle\psi_{G^k}|G^k|\psi_G\rangle \\ &= \langle(G^k)^\dagger\psi_{G^k}|\psi_G\rangle. \end{aligned}$$

Usando o fato de que

$$(G^k)^\dagger|\psi_{G^k}\rangle = (G^k)^\dagger G^k|\psi\rangle = |\psi\rangle,$$

obtemos, para todo $k \in \mathbb{N}$,

$$\begin{aligned}\langle\psi_{G^k}|\psi_{G^{k+1}}\rangle &= \langle\psi|\psi_G\rangle \\ &= \cos\theta,\end{aligned}$$

como queríamos demonstrar. ■

PROPOSIÇÃO 3 *O operador G rotaciona $|\psi\rangle$ em direção a $|i_0\rangle$.*

Prova. Inicialmente, calculemos o ângulo θ entre os vetores $|\psi\rangle$ e $G|\psi\rangle$. De (3.16) e (3.23), temos:

$$\begin{aligned}\cos\theta &= \langle\psi|\psi_G\rangle \\ &= \frac{N-4}{N}\langle\psi|\psi\rangle + \frac{2}{\sqrt{N}}\langle\psi|i_0\rangle \\ &= \frac{N-4}{N} + \frac{2}{\sqrt{N}}\left(\frac{1}{\sqrt{N}}\right) \\ &= \frac{N-2}{N}.\end{aligned} \tag{3.31}$$

Calculemos, agora, o ângulo entre $G|\psi\rangle$ e $|i_0\rangle$. De (3.16) e (3.25), temos:

$$\begin{aligned}\langle\psi_G|i_0\rangle &= \frac{N-4}{N}\langle\psi|i_0\rangle + \frac{2}{\sqrt{N}}\langle i_0|i_0\rangle \\ &= \frac{N-4}{N\sqrt{N}} + \frac{2}{\sqrt{N}} \\ &= \frac{3N-4}{N\sqrt{N}}.\end{aligned}$$

Para uma lista com 2 elementos ($N = 2$), o algoritmo de Grover "não funciona" (dê uma justificativa para isso). Vamos supor, então, que $N > 2$. Neste caso,

$$\frac{3N-4}{N\sqrt{N}} > \frac{1}{\sqrt{N}},$$

ou melhor,

$$\langle\psi_G|i_0\rangle > \langle\psi|i_0\rangle.$$

Como a função *arccos* é decrescente no intervalo $[-1, 1]$, a desigualdade acima é equivalente a

$$\arccos(\langle\psi_G|i_0\rangle) < \arccos(\langle\psi|i_0\rangle).$$

Da Proposição 1, $|\psi_G\rangle \in \Omega$ e, de (3.31), sabemos que a rotação produzida por G é, no máximo, de $\pi/2$ rad. Portanto, usando a desigualdade acima, a única possibilidade é que a rotação de $|\psi\rangle$ seja em direção a $|i_0\rangle$. ■

PROPOSIÇÃO 4 *A aplicação de G sobre $|\psi_{G^n}\rangle$, para todo $n \in \mathbb{N}$, mantém o mesmo sentido de rotação quando G é aplicado sobre $|\psi\rangle$.*

Prova. Pelas Proposições 1, 2 e 3, já sabemos que, quando aplicamos o operador G sobre o estado $|\psi_{G^n}\rangle$, temos apenas duas possibilidades: $G\left(G^n|\psi\rangle\right)$ é um estado resultante de uma rotação de θ rad, em Ω, no sentido horário ou anti-horário. Se demonstrarmos que, para todo $n \in \mathbb{N}$,

$$G\left(G^n|\psi\rangle\right) \neq G^{n-1}|\psi\rangle,$$

poderemos concluir que a rotação mantém o mesmo sentido quando G é aplicado sobre $|\psi\rangle$. A demonstração será, portanto, por indução. Inicialmente, mostremos que

$$G\left(G^1|\psi\rangle\right) \neq G^0|\psi\rangle,$$

ou seja,

$$G|\psi_G\rangle \neq |\psi\rangle.$$

Usando (3.25) e (3.30), podemos calcular $G|\psi_G\rangle$:

$$\begin{aligned}
G|\psi_G\rangle &= G\left(\frac{N-4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle\right) \\
&= \frac{N-4}{N}G|\psi\rangle + \frac{2}{\sqrt{N}}G|i_0\rangle \\
&= \frac{N-4}{N}\left(\frac{N-4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle\right) + \frac{2}{\sqrt{N}}\left(-\frac{2}{\sqrt{N}}|\psi\rangle + |i_0\rangle\right) \\
&= \left(\frac{N-4}{N}\right)^2|\psi\rangle + \frac{2N-8}{N\sqrt{N}}|i_0\rangle - \frac{4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle \\
&= \left(\left(\frac{N-4}{N}\right)^2 - \frac{4}{N}\right)|\psi\rangle + \frac{4N-8}{N\sqrt{N}}|i_0\rangle.
\end{aligned}$$

Para $N > 2$, este estado é diferente de $|\psi\rangle$. Suponhamos agora que, para um dado $k \in \mathbb{N}$,

$$G\left(G^k|\psi\rangle\right) \neq G^{k-1}|\psi\rangle.$$

Como G é um operador unitário, podemos aplicá-lo nos dois lados da expressão acima e ainda obter estados distintos, isto é,

$$G\left(G^{k+1}|\psi\rangle\right) \neq G^k|\psi\rangle.$$

Isso conclui a indução (dê um exemplo mostrando que a conclusão da indução só é possível, porque G é um operador unitário). ■

Conclusão: a aplicação de G^k sobre $|\psi\rangle$ produz uma rotação de $k\theta$ rad em direção a $|i_0\rangle$, no subespaço gerado por $|\psi\rangle$ e $|i_0\rangle$, para todo $k \in \mathbb{N}$.

Consideremos, então, o "custo" do algoritmo de Grover. De forma mais precisa, devemos calcular o número de vezes k que o operador G deve ser aplicado para

que o estado $G^k|\psi\rangle$ torne-se o mais próximo do estado $|i_0\rangle$. Dito de outra forma, queremos saber que valor de k faz com que o ângulo entre $|i_0\rangle$ e $G^k|\psi\rangle$ seja o mais próximo de zero (Figura 3.8). Admitindo que k seja um número real, podemos representar matematicamente o problema acima através da seguinte equação:

$$\arccos(\langle\psi|i_0\rangle) - k\theta = 0. \tag{3.32}$$

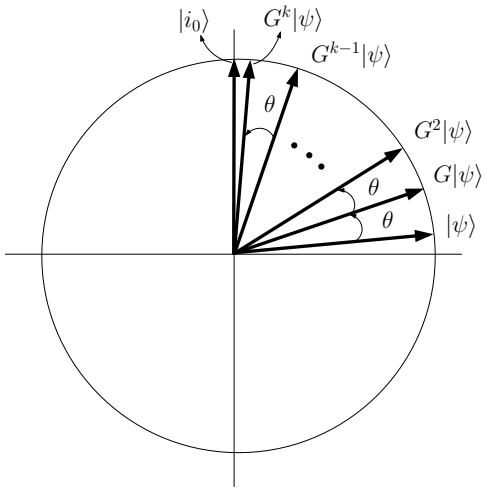

Figura 3.8: Aplicações sucessivas do operador G.

De (3.31), já sabemos que o ângulo θ entre $|\psi\rangle$ e $G|\psi\rangle$ é

$$\theta = \arccos\left(\frac{N-2}{N}\right). \tag{3.33}$$

Substituindo (3.16) e (3.33) em (3.32), obtemos:

$$\arccos\left(\frac{1}{\sqrt{N}}\right) - k\arccos\left(\frac{N-2}{N}\right) = 0.$$

Isolando k, temos:

$$k = \frac{\arccos\left(\frac{1}{\sqrt{N}}\right)}{\arccos\left(\frac{N-2}{N}\right)}. \tag{3.34}$$

Para sabermos a ordem de grandeza de k, inicialmente, "comparemos" k com N. Calculando o limite, temos:

$$\lim_{N\to\infty}\frac{k}{N} = 0.$$

Ou seja, k é "menor" do que N, para valores grandes de N. Calculemos, então, o seguinte:

$$\lim_{N\to\infty} \frac{k}{\log_2(N)} = \infty.$$

Neste caso, k é "maior" do que $\log_2(N)$, para valores grandes de N. Tentando um valor "intermediário", obtemos:

$$\lim_{N\to\infty} \frac{k}{\sqrt{N}} = \frac{\pi}{4}.$$

Isso significa que, para valores suficientemente grandes de N, o número de vezes que o operador G deve ser aplicado é, no máximo, $\sqrt{N}$ vezes.

Esse é o resultado que tínhamos enunciado no início do capítulo. Na próxima seção, daremos um exemplo usando uma lista com 8 elementos.

Exercício 3.6 Calcule os três limites acima.

3.4 Exemplo: N=8

Apliquemos o algoritmo de Grover em uma lista com $N = 8$ elementos. O primeiro registrador terá, portanto, 3 q-bits. A primeira pergunta é: quantas aplicações do operador G devem ser utilizadas? Usando (3.34), obtemos:

$$k = \frac{\arccos\left(\frac{1}{\sqrt{8}}\right)}{\arccos\left(\frac{8-2}{8}\right)} \cong 1,67.$$

Para que o estado resultante da última aplicação de G esteja o mais próximo de $|i_0\rangle$, devemos aplicar 2 vezes o operador G (Figura 3.9). A idéia é arredondar o valor de k para o inteiro mais próximo.

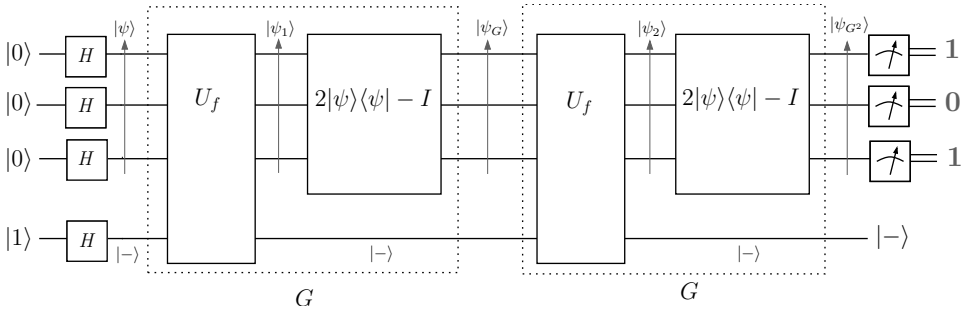

Figura 3.9: Duas aplicações do operador G, para $N = 8$ e $i_0 = 101$.

Antes da aplicação de G, o algoritmo cria uma superposição $|\psi\rangle$ formada por todos os elementos da base computacional associada ao problema. Isso é obtido aplicando o operador H (2.2) sobre cada q-bit do estado inicial ($|000\rangle$) do primeiro registrador, isto é,

$$\begin{aligned}
|\psi\rangle &= H|0\rangle \otimes H|0\rangle \otimes H|0\rangle \\
&= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \otimes \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right) \\
&= \frac{1}{\sqrt{8}}\left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\right).
\end{aligned}$$

Em notação decimal, temos:

$$|\psi\rangle = \frac{1}{\sqrt{8}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle\right).$$

Supondo que o elemento procurado seja

$$|i_0\rangle = |101\rangle = |5\rangle,$$

o próximo passo é aplicar o operador U_f sobre o estado $|\psi\rangle|-\rangle$. O elemento procurado é, então, o único que tem sua amplitude alterada:

$$\begin{aligned}
|\psi_1\rangle|-\rangle &= U_f\left(|\psi\rangle|-\rangle\right) \\
&= \left(\frac{|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle - |5\rangle + |6\rangle + |7\rangle}{\sqrt{8}}\right)|-\rangle.
\end{aligned}$$

Em seguida, o operador $2|\psi\rangle\langle\psi| - I$ é aplicado sobre o estado $|\psi_1\rangle$, produzindo o estado $|\psi_G\rangle$:

$$\begin{aligned}
|\psi_G\rangle &= \left(2|\psi\rangle\langle\psi| - I\right)|\psi_1\rangle \\
&= \left(2\langle\psi|\psi_1\rangle\right)|\psi\rangle - |\psi_1\rangle \\
&= \frac{3}{2}|\psi\rangle - |\psi_1\rangle \\
&= \frac{1}{2\sqrt{8}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |6\rangle + |7\rangle\right) + \frac{5}{2\sqrt{8}}|5\rangle.
\end{aligned}$$

Se medirmos este estado, a probabilidade de se obter o elemento procurado é

$$\left(\frac{5}{2\sqrt{8}}\right)^2 \cong 78,12\%.$$

Entretanto, já sabemos que devemos aplicar 2 vezes o operador G. Aplicando o operador U_f sobre o estado $|\psi_G\rangle|-\rangle$, obtemos:

$$\begin{aligned}
|\psi_2\rangle|-\rangle &= U_f\left(|\psi_G\rangle|-\rangle\right) \\
&= \left(\frac{1}{2\sqrt{8}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |6\rangle + |7\rangle\right) - \frac{5}{2\sqrt{8}}|5\rangle\right)|-\rangle.
\end{aligned}$$

Novamente, o elemento procurado é o único que tem sua amplitude alterada. Aplicando o operador $2|\psi\rangle\langle\psi| - I$ sobre $|\psi_2\rangle$, temos:

$$\begin{aligned}
|\psi_{G^2}\rangle &= (2|\psi\rangle\langle\psi| - I)\,|\psi_2\rangle \\
&= (2\langle\psi|\psi_2\rangle)\,|\psi\rangle - |\psi_2\rangle \\
&= \frac{1}{4}|\psi\rangle - |\psi_2\rangle \\
&= \left(\frac{-1}{4\sqrt{8}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |6\rangle + |7\rangle\right) + \frac{11}{4\sqrt{8}}|5\rangle\right)|-\rangle.
\end{aligned}$$

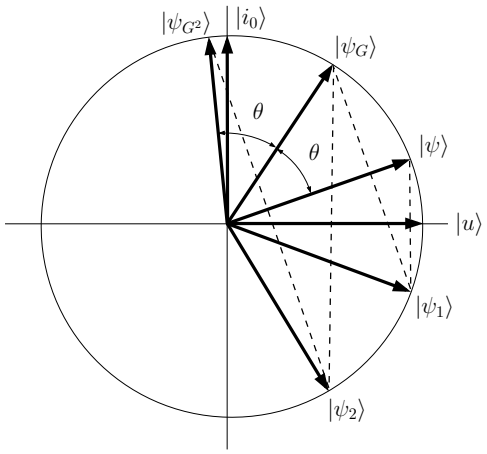

Figura 3.10: Duas aplicações do operador G, para $N = 8$ e $i_0 = 101$.

Fazendo uma medida do estado $|\psi_{G^2}\rangle$, obtemos o elemento procurado com probabilidade de

$$\left(\frac{11}{4\sqrt{8}}\right)^2 \cong 94,53\%.$$

Na Figura 3.10, representamos geometricamente os passos do algoritmo resultantes de duas aplicações do operador G.

Exercício 3.7 Usando a Figura 3.10, dê uma explicação para os sinais das amplitudes da superposição dada em $|\psi_{G^2}\rangle$.

3.5 Circuitos Quânticos para o Operador G

Nesta seção, iremos decompor o operador G em termos de portas de 1 q-bit e portas CNOT. Essa decomposição mostrará como poderia ser uma implementação prática do operador G.

3.5.1 Circuito quântico para o operador U_f

Recordemos que a função f (3.1) age como um oráculo para identificar o elemento procurado i_0. De forma similar, o operador U_f também pode ser imaginado como um oráculo. Nesse sentido, ele é um operador diferente do operador $2|\psi\rangle\langle\psi| - I$, pois deve ser "preparado" para a identificação do estado $|i_0\rangle$. O operador U_f pode ser representado por uma porta Toffoli generalizada com n q-bits de controle, 1 q-bit alvo no estado $|-\rangle$ e 2 portas X atuando no i-ésimo q-bit de controle, sempre que o i-ésimo dígito binário de i_0 for 0. Por exemplo, o circuito quântico para o operador U_f, usado no exemplo dado na Seção 3.4 ($n = 3$ e $i_0 = 101$), tem a forma apresentada na Figura 3.11. Se o elemento procurado fosse 111, nenhuma porta X seria usada. Caso fosse 000, 3 pares de portas X seriam usadas, um par em cada q-bit de controle.

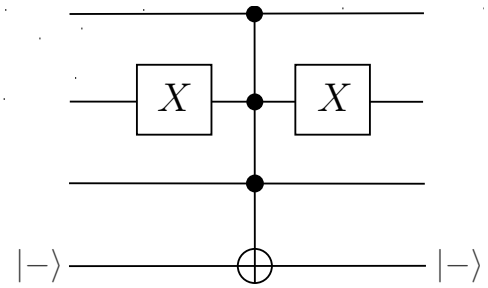

Figura 3.11: Circuito quântico para o operador U_f ($n = 3$ e $i_0 = 101$).

3.5.2 Circuito quântico para o operador $2|\psi\rangle\langle\psi| - I$

Consideremos, agora, a decomposição do operador $2|\psi\rangle\langle\psi| - I$. Usando

$$|\psi\rangle = H^{\otimes n}|0\rangle \text{ e } \langle\psi| = \langle 0|(H^{\otimes n})^\dagger,$$

temos, então,

$$\begin{aligned} 2|\psi\rangle\langle\psi| - I &= 2H^{\otimes n}(|0\rangle\langle 0|)(H^{\otimes n})^\dagger - I \\ &= H^{\otimes n}(2|0\rangle\langle 0|)(H^{\otimes n})^\dagger - H^{\otimes n}(H^{\otimes n})^\dagger \\ &= H^{\otimes n}(2|0\rangle\langle 0| - I)(H^{\otimes n})^\dagger \\ &= H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}. \end{aligned} \tag{3.35}$$

Observe que $H^{\otimes n}$ é uma matriz simétrica com apenas entradas reais. Portanto, $(H^{\otimes n})^\dagger = H^{\otimes n}$.

Exercício 3.8 Demonstre que o produto tensorial de matrizes simétricas é uma matriz simétrica.

A equação (3.35) mostra que, para obtermos o circuito quântico do operador $2|\psi\rangle\langle\psi| - I$, basta considerarmos o operador $2|0\rangle\langle 0| - I$. Esse operador faz uma reflexão em relação ao estado $|0\rangle$. O circuito para esse operador é dado na Figura 3.12. Na Tabela 3.1, representamos a ação desse operador sobre o estado $|0\rangle$.

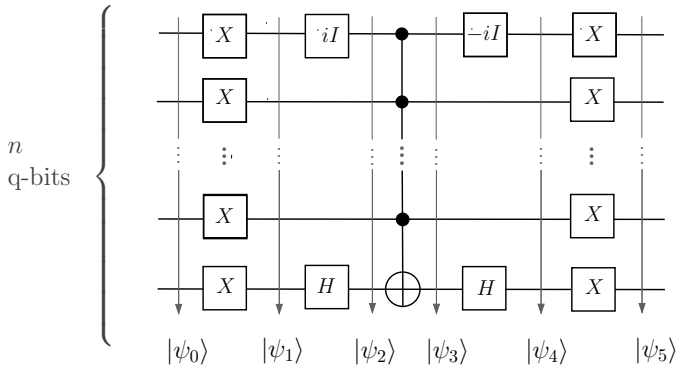

Figura 3.12: Circuito quântico para o operador $2|0\rangle\langle 0| - I$.

Observe que a única porta que atua nos n q-bits ao mesmo tempo, na Figura 3.12, é a porta Toffoli generalizada (Figura 2.5).

Exercício 3.9 Teste a ação do circuito da Figura 3.12 em outros estados da base computacional para perceber que, para qualquer entrada $|j\rangle$, com $0 < j < N$, a saída será sempre -$|j\rangle$.

$\|\psi_0\rangle$	$\|\psi_1\rangle$	$\|\psi_2\rangle$	$\|\psi_3\rangle$	$\|\psi_4\rangle$	$\|\psi_5\rangle$
$\|0\rangle$	$\|1\rangle$	$i\|1\rangle$	$i\|1\rangle$	$(-i \cdot i)\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
⋮	⋮	⋮	⋮	⋮	⋮
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|-\rangle$	-$\|-\rangle$	$\|1\rangle$	$\|0\rangle$

Tabela 3.1: Ação do operador $2|0\rangle\langle 0| - I$ no estado $|0\rangle$ da base computacional.

Capítulo 4

Algoritmo de Shor

4.1 Redução da Fatoração ao Cálculo de Ordem

Neste capítulo[1], vamos descrever o algoritmo de Shor [21] que acha os fatores primos de um número composto N. Começaremos mostrando como a fatoração pode ser reduzida ao cálculo da ordem de um número x menor que N, escolhido aleatoriamente. Imagine que N é um número grande, por exemplo com 300 dígitos na notação decimal, já que tais números são usados em criptografia. Embora N seja grande, o número de q-bits necessário para guardá-lo é muito pequeno. Em geral, $\log_2 N$ não é um inteiro, então definimos

$$n = \lceil \log_2 N \rceil,$$

onde $\lceil \log_2 N \rceil$ é o menor inteiro maior ou igual a $\log_2 N$.

Um computador quântico com n q-bits pode guardar N ou qualquer outro inteiro positivo menor que N. Facilmente, vemos que o número de fatores primos de N é no máximo n. Se o número de q-bits e o número de fatores primos são menores ou iguais a n, então é natural perguntar se existe um algoritmo que fatora N em um número de passos que é polinomial em n. Mais precisamente a questão é: existe um algoritmo de fatoração na classe de complexidade $\mathcal{P}$ [17]?

A redução da fatoração de N ao problema de achar a *ordem* de um inteiro x menor que N pode ser descrita da seguinte forma. Se x e N possuem fatores comuns, então o $\mathrm{MDC}(x, N)$ fornece um fator de N; portanto, é suficiente investigar o caso quando x é coprimo com N. A ordem de x, módulo N, é o menor inteiro positivo r, tal que

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

Se r for par, podemos definir y como sendo

$$x^{r/2} \equiv y \bmod N.$$

[1] Os exercícios deste capítulo estão embutidos no próprio texto, com exceção dos três últimos na página 58.

A notação acima significa que y é o resto da divisão de $x^{r/2}$ por N e, pela definição, $0 \leq y < N$. Note que y satisfaz $y^2 \equiv 1 \bmod N$, ou equivalentemente, $(y-1)(y+1) \equiv 0 \bmod N$, o que significa que N divide $(y-1)(y+1)$. Se $1 < y < N-1$, os fatores $y-1$ e $y+1$ satisfazem $0 < y-1 < y+1 < N$; portanto, N não pode dividir $y-1$ nem $y+1$ separadamente. A única alternativa é que ambos $y-1$ e $y+1$ tenham fatores de N. Então, MDC$(y-1, N)$ e MDC$(y+1, N)$ produzem fatores não triviais de N. Se N tiver mais fatores, eles podem ser calculados aplicando o algoritmo recursivamente. Considere $N = 21$ como exemplo. A seqüência de equivalências

$$\begin{aligned} 2^4 &\equiv 16 \bmod 21 \\ 2^5 &\equiv 11 \bmod 21 \\ 2^6 &\equiv 11 \times 2 \equiv 1 \bmod 21 \end{aligned}$$

mostra que a ordem de 2, módulo 21, é $r = 6$. Portanto, $y \equiv 2^3 \equiv 8 \bmod 21$. De $y-1$ resulta o fator 7 e de $y+1$ resulta o fator 3 de 21. Resumindo, se escolhermos aleatoriamente um inteiro positivo x menor que N e calcularmos o MDC(x, N), ou teremos um fator de N, ou ficaremos sabendo que x é coprimo com N. Neste último caso, se x satisfizer as condições (1) a ordem r é par, e (2) $0 < y-1 < y+1 < N$, então o MDC$(y-1, N)$ e MDC$(y+1, N)$ produzem fatores de N. Se uma das condições não é verdadeira, recomeçamos até achar um candidato x apropriado. O método não seria útil se estas suposições fossem restritivas demais, mas felizmente este não é o caso. O método sistematicamente falha, se N for uma potência de algum primo, mas nesse caso, um algoritmo clássico eficiente é conhecido. Se N for par, podemos continuar dividindo por 2 até o resultado passar a ser ímpar. Resta-nos aplicar o método para os inteiros compostos ímpares que não são potências de algum número primo. É complicado provar que a probabilidade de achar x coprimo com N satisfazendo as condições (1) e (2) é alta; de fato, essa probabilidade é $1 - 1/2^{k-1}$, onde k é o número de fatores primos de N. No pior dos casos (N tem 2 fatores), a probabilidade é maior ou igual a $1/2$ (veja a prova no Apêndice B de [10]). À primeira vista, parece que acabamos de descrever um algoritmo eficiente para achar um fator de N. Isto não é verdade, já que não são conhecidos algoritmos clássicos eficientes para calcular a ordem de um inteiro x módulo N. Por outro lado, existe (depois do trabalho de Shor) um algoritmo quântico eficiente. Vamos descrevê-lo a seguir.

4.2 Algoritmo Quântico para o Cálculo de Ordem

Considere o circuito da Figura 4.1 que calcula a ordem r de um inteiro positivo x menor que N, coprimo com N. V_x é um operador linear unitário dado por

$$V_x(|j\rangle\,|k\rangle) = |j\rangle\,\left|k + x^j\right\rangle, \tag{4.1}$$

onde $|j\rangle$ e $|k\rangle$ são os estados do primeiro e segundo registrador, respectivamente. As operações aritméticas são feitas módulo N, assim $0 \leq k + x^j < N$. O operador DFT (*Discrete Fourier Transform*) será descrito mais adiante.

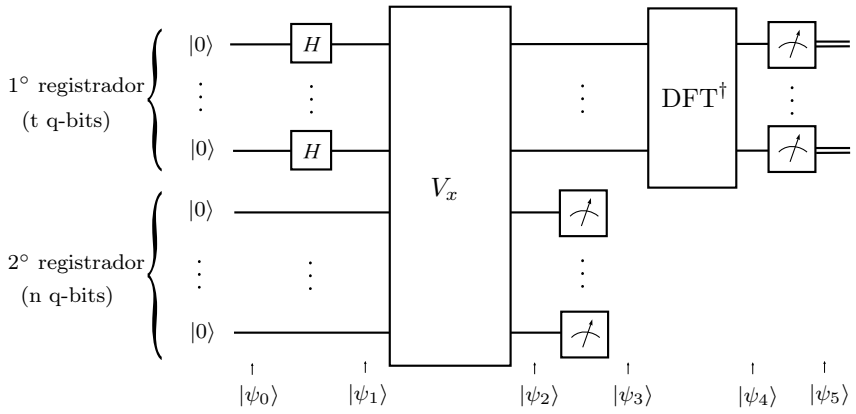

Figura 4.1: Circuito quântico para achar a ordem de um inteiro positivo x módulo N.

O primeiro registrador possui t q-bits, onde t deve ser escolhido de forma que $N^2 \leq 2^t \leq 2N^2$, por razões que ficarão claras mais adiante [20]. Se a ordem é uma potência de 2, então é suficiente tomar $t = n$. Nesta seção, vamos considerar este caso especial e deixar o caso geral para a Seção 4.4. Continuaremos usando a variável t para generalizarmos nossa discussão mais adiante.

Os estados do computador quântico estão indicados por $|\psi_0\rangle$ até $|\psi_5\rangle$ na Figura 4.1. O estado inicial é

$$|\psi_0\rangle = \underbrace{|0\ldots0\rangle}_{t}\underbrace{|0\ldots0\rangle}_{n}.$$

A aplicação do operador Hadamard

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

em cada q-bit do primeiro registrador, resulta em

$$|\psi_1\rangle = \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle\,|0\rangle\,. \tag{4.2}$$

O primeiro registrador está em uma superposição de todos os estados da base computacional com igual amplitude dada por $\frac{1}{\sqrt{2^t}}$. Agora, note o que acontece quando

aplicamos V_x em $|\psi_1\rangle$:

$$\begin{aligned} |\psi_2\rangle &= V_x |\psi_1\rangle \\ &= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} V_x(|j\rangle |0\rangle) \\ &= \frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} |j\rangle \left|x^j\right\rangle . \end{aligned} \tag{4.3}$$

O estado $|\psi_2\rangle$ é interessante, pois, já que V_x é linear, ele atua simultaneamente em todos os termos $|j\rangle |0\rangle$ para 2^t valores de j. Logo, isto gera todas as potências de x simultaneamente. Esta característica é chamada *paralelismo quântico*. Algumas dessas potências são 1, as quais correspondem aos estados

$$|0\rangle |1\rangle , |r\rangle |1\rangle , |2r\rangle |1\rangle , \cdots , \left|\left(\frac{2^t}{r} - 1\right) r\right\rangle |1\rangle . \tag{4.4}$$

Isto explica a escolha de (4.1) para V_x. Classicamente, poderíamos calcular sucessivamente x^j, para j começando de 2 até chegarmos a $j = r$. Quanticamente, pode-se calcular todas as potências de x com uma única aplicação de V_x. No nível quântico, todos os valores de j que produzem $x^j \equiv 1 \bmod N$ são "conhecidos". Mas esta informação não está totalmente disponível no nível clássico. Uma informação clássica de um estado quântico é obtida através de uma medida e, neste ponto, não ajudaria se medíssemos o primeiro registrador, já que todos os estados da superposição (4.3) possuem igual amplitude. A primeira parte da estratégia para determinar r é observar que o primeiro registrador dos estados (4.4) é periódico. Então, a informação que queremos é um período.

Para facilitar os cálculos, vamos medir o segundo registrador. Antes de fazer isto, vamos reescrever o estado $|\psi_2\rangle$, fatorando os termos iguais do segundo registrador. Como x^j é uma função periódica com período r, vamos substituir j por $ar + b$ na equação (4.3), onde $0 \leq a \leq (2^t/r) - 1$ e $0 \leq b \leq r - 1$. Lembre-se que supomos que $t = n$ e r é uma potência de 2, portanto r divide 2^t. A equação (4.3) é convertida em

$$|\psi_2\rangle = \frac{1}{\sqrt{2^t}} \sum_{b=0}^{r-1} \left(\sum_{a=0}^{\frac{2^t}{r}-1} |ar + b\rangle \right) \left|x^b\right\rangle . \tag{4.5}$$

No segundo registrador, substituímos x^b por x^{ar+b}, já que $x^r \equiv 1 \bmod N$. Agora, o segundo registrador é medido. Qualquer resultado x^0, x^1, ..., x^{r-1} pode ser obtido com igual probabilidade. Suponha que o resultado é x^{b_0}. O estado do computador quântico é agora

$$|\psi_3\rangle = \sqrt{\frac{r}{2^t}} \left(\sum_{a=0}^{\frac{2^t}{r}-1} |ar + b_0\rangle \right) \left|x^{b_0}\right\rangle . \tag{4.6}$$

Note que depois da medida, a constante é renormalizada para $\sqrt{r/2^t}$, já que existem $2^t/r$ termos na soma (4.6). A Figura 4.2 mostra a probabilidade de obtermos os estados da base computacional, medindo o primeiro registrador. A probabilidade forma uma função periódica com período r. Estes valores são zero, exceto para os estados $|b_0\rangle$, $|r + b_0\rangle$, $|2r + b_0\rangle$, ..., $|2^t - r + b_0\rangle$.

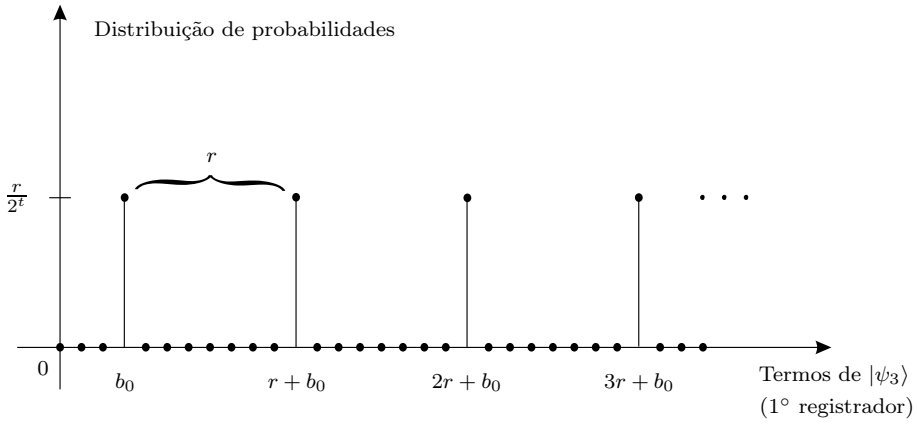

Figura 4.2: Distribuição de probabilidades de $|\psi_3\rangle$ medido na base computacional (para o caso $b_0 = 3$ and $r = 8$). O eixo horizontal tem 2^t pontos, o número de picos é $2^t/r$ e o período é r.

Como podemos descobrir o período de uma função eficientemente? A resposta é a transformada de Fourier. A transformada de Fourier de uma função periódica de período r é uma nova função com período proporcional a $1/r$. Isto faz diferença para determinar r. A transformada de Fourier é a segunda e última parte da estratégia usada por Shor. Todo o método depende de um algoritmo quântico eficiente para calcular a transformada de Fourier, que não está disponível classicamente. Na Seção 4.5, mostraremos que a transformada de Fourier é calculada eficientemente num computador quântico.

4.3 A Transformada de Fourier Quântica Discreta

A transformada de Fourier de uma função $F : \{0, \ldots, N-1\} \to \mathbb{C}$ é uma nova função $\tilde{F} : \{0, \ldots, N-1\} \to \mathbb{C}$ definida por

$$\tilde{F}(k) = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k/N} F(j). \tag{4.7}$$

Podemos aplicar a transformada de Fourier ou em uma função ou em um estado da base computacional. A transformada de Fourier aplicada ao estado $|k\rangle$ da base

computacional $\{|0\rangle, \ldots, |N-1\rangle\}$ é

$$\mathrm{DFT}(|k\rangle) = |\psi_k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i j k/N} |j\rangle , \tag{4.8}$$

onde o conjunto $\{|\psi_k\rangle : k = 0, \ldots, N-1\}$ forma uma nova base ortonormal. A transformada de Fourier é um operador linear unitário; portanto, se sabemos como ele atua nos estados da base computacional, também sabemos como ele atua num estado genérico

$$|\psi\rangle = \sum_{a=0}^{N-1} F(a) |a\rangle .$$

A transformada de Fourier de $|\psi\rangle$ pode ser obtida através de (4.7) ou (4.8). No resto do trabalho, vamos usar a última forma.

Para provar que $\{|\psi_k\rangle : k = 0, \ldots, N-1\}$ é uma base ortonormal, i.e.,

$$\langle \psi_{k'} | \psi_k \rangle = \delta_{k'k},$$

podemos usar a identidade

$$\frac{1}{N} \sum_{j=0}^{N-1} e^{2\pi i j k/N} = \begin{cases} 1, \text{ se } k \text{ é um múltiplo de } N \\ 0, \text{ caso contrário,} \end{cases} \tag{4.9}$$

que é útil no contexto da transformada de Fourier. É fácil verificar que (4.9) é verdadeira. Se k é um múltiplo de N, então $e^{2\pi i j k/N} = 1$, justificando o primeiro caso da identidade. Se k não é um múltiplo de N, (4.9) é verdade, mesmo se N não for uma potência de 2. A Figura 4.3 mostra cada termo $e^{2\pi i j k/N}$ $(j = 0, ..., 6)$ para o caso $k = 1$ e $N = 7$ como vetores num plano complexo. Note que a soma dos vetores deve ser zero pelo argumento de simetria: a distribuição dos vetores é isotrópica. Usualmente, diz-se que a interferência é destrutiva neste caso. Usando essa identidade, podemos definir a transformada de Fourier inversa, que é similar a (4.8), porém com um sinal de menos no expoente. Note que $\mathrm{DFT}^{-1} = \mathrm{DFT}^{\dagger}$, já que DFT é um operador unitário.

Apresentaremos os detalhes de um circuito quântico para realizar a transformada de Fourier na Seção 4.5. Agora, continuaremos o processo de cálculo da Figura 4.1.

Estamos prontos para achar o próximo estado do computador quântico: $|\psi_4\rangle$. Aplicando a transformada de Fourier inversa no primeiro registrador, usando a equação (4.8) e a linearidade da $\mathrm{DFT}^{\dagger}$, obtemos

$$\begin{aligned} |\psi_4\rangle &= \mathrm{DFT}^{\dagger}(|\psi_3\rangle) \\ &= \sqrt{\frac{r}{2^t}} \sum_{a=0}^{\frac{2^t}{r}-1} \left(\frac{1}{\sqrt{2^t}} \sum_{j=0}^{2^t-1} e^{-2\pi i j (ar+b_0)/2^t} |j\rangle \right) |x^{b_0}\rangle . \end{aligned}$$

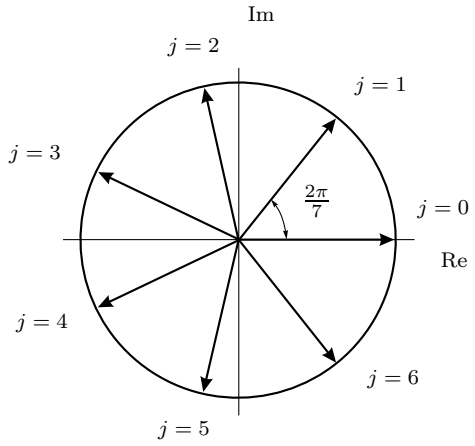

Figura 4.3: Desenho dos vetores $e^{2\pi ij/7}$ $(j = 0, ..., 6)$ no plano complexo. A soma desses vetores é zero por argumentos de simetria. Este é um exemplo da equação (4.9) para $N = 7$, $k = 1$.

Invertendo a ordem do somatório, temos

$$|\psi_4\rangle = \frac{1}{\sqrt{r}} \left(\sum_{j=0}^{2^t-1} \left[\frac{1}{2^t/r} \sum_{a=0}^{\frac{2^t}{r}-1} e^{\frac{-2\pi ija}{2^t/r}} \right] e^{-2\pi ijb_0/2^t} |j\rangle \right) \left|x^{b_0}\right\rangle . \tag{4.10}$$

Usando (4.9), vemos que a expressão nos colchetes é diferente de zero, se e somente se $j = k2^t/r$ com $k = 0, ..., r-1$. Quando j assume tais valores, a expressão nos colchetes é igual a 1. Então, temos

$$|\psi_4\rangle = \frac{1}{\sqrt{r}} \left(\sum_{k=0}^{r-1} e^{-2\pi i \frac{k}{r} b_0} \left| \frac{k2^t}{r} \right\rangle \right) \left|x^{b_0}\right\rangle . \tag{4.11}$$

Para acharmos r, a expressão $|\psi_4\rangle$ tem duas vantagens sobre a expressão $|\psi_3\rangle$ (equação (4.6)): r está no denominador do *ket* e o parâmetro aleatório b_0 foi movido do *ket* para o expoente, ocupando agora um lugar "inofensivo". A Figura 4.4 mostra a distribuição de probabilidades de $|\psi_4\rangle$ medido na base computacional. Medindo o primeiro registrador, obtemos o valor $k_0 2^t/r$, onde k_0 pode ser qualquer número entre 0 e $r-1$, com igual probabilidade (picos na Figura 4.4). Se obtivermos $k_0 = 0$, não teremos nenhuma informação sobre r, e o algoritmo tem que ser rodado novamente. Se $k_0 \neq 0$, dividimos $k_0 2^t/r$ por 2^t, obtendo k_0/r. Nem k_0 nem r são conhecidos. Se k_0 é coprimo com r, simplesmente selecionamos o denominador.

Se k_0 e r têm fator comum, o denominador da fração reduzida k_0/r é um fator de r, mas não é o próprio r. Suponha que o denominador é r_1. Seja $r = r_1 r_2$. Agora, o objetivo é determinar r_2, que é a ordem de x^{r_1}, módulo N. Rodamos novamente a parte quântica do algoritmo para achar a ordem de x^{r_1}. Se acharmos r_2 na primeira

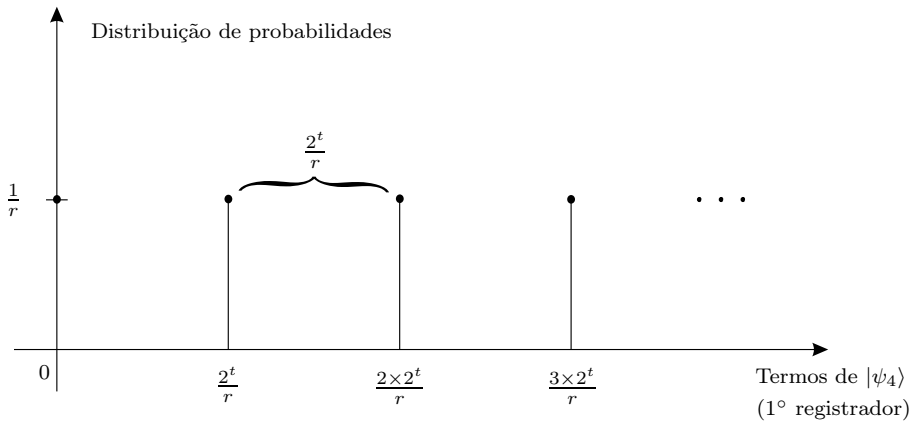

Figura 4.4: Distribuição de probabilidades de $|\psi_4\rangle$ medida na base computacional. O eixo horizontal tem 2^t pontos; apenas os termos não nulos são mostrados. O número de picos é r e o período é $2^t/r$.

rodada, o algoritmo pára; caso contrário, o aplicamos recursivamente. O processo recursivo não é longo, porque o número de iterações é menor ou igual a $\log_2 r$.

Como exemplo, tome $N = 15$ que é o menor número composto não trivial. O conjunto de números menores que 15 e coprimos com 15 é $\{1, 2, 4, 7, 8, 11, 13, 14\}$. Os elementos 4, 11 e 14 têm ordem 2 e os elementos 2, 7, 8 e 13 têm ordem 4. Portanto, em qualquer caso, r é uma potência de 2 e os fatores de $N = 15$ podem ser encontrados num computador quântico com $\lceil \log_2 15 \rceil = 8$ q-bits. Os autores de [23] usam um computador quântico com 7 q-bits, pulando partes do algoritmo original.

4.4 Generalização por meio de um Exemplo

Nas seções precedentes, consideramos um caso especial quando a ordem r é uma potência de 2 e $t = n$ (t é o número de q-bits no primeiro registrador, veja na Figura 4.1, e $n = \lceil \log_2 N \rceil$). Nesta seção, consideramos a fatoração de $N = 21$, que é o próximo número composto não trivial depois de $N = 15$. Devemos escolher t tal que 2^t esteja entre N^2 e $2N^2$, o que é sempre possível [20]. Para $N = 21$, o menor valor de t é 9. Este é o exemplo mais simples permitido pelos vínculos, mas é suficiente para mostrar todas as propriedades do algoritmo de Shor. O primeiro passo é escolher x aleatoriamente, tal que $1 < x < N$, e testar se x é coprimo com N. Se não for, encontramos facilmente um fator de N pelo cálculo do MDC(x, N). Se for, iniciamos a parte quântica do algoritmo. Suponha que $x = 2$ foi escolhido. O objetivo é encontrar a ordem de x, que é $r = 6$. O computador quântico é inicializado no estado

$$|\psi_0\rangle = |0\rangle\,|0\rangle\,,$$

onde o primeiro registrador tem $t = 9$ q-bits e o segundo tem $n = 5$ q-bits. O próximo passo é a aplicação de $H^{\otimes 9}$ sobre o primeiro registrador, gerando (veja equação (4.2))

$$|\psi_1\rangle = \frac{1}{\sqrt{512}} \sum_{j=0}^{511} |j\rangle\,|0\rangle\,.$$

Em seguida, aplicando V_x (definido em (4.1)), obtemos

$$\begin{aligned}
|\psi_2\rangle &= \frac{1}{\sqrt{512}} \sum_{j=0}^{511} |j\rangle\,\big|2^j \bmod N\big\rangle \\
&= \frac{1}{\sqrt{512}}\Big(\quad |0\rangle\,|1\rangle + |1\rangle\,|2\rangle + |2\rangle\,|4\rangle + |3\rangle\,|8\rangle + \quad |4\rangle\,|16\rangle + \quad |5\rangle\,|11\rangle + \\
&\qquad\qquad |6\rangle\,|1\rangle + |7\rangle\,|2\rangle + |8\rangle\,|4\rangle + |9\rangle\,|8\rangle + |10\rangle\,|16\rangle + |11\rangle\,|11\rangle + \\
&\qquad\qquad |12\rangle\,|1\rangle + \ldots \Big).
\end{aligned}$$

Note que a expressão acima tem o seguinte padrão: o estado do segundo registrador de cada "coluna" é o mesmo. Portanto, podemos rearranjar os termos de forma a fatorar o segundo registrador:

$$\begin{aligned}
|\psi_2\rangle = \frac{1}{\sqrt{512}}\Big[&\big(\ |0\rangle + \ |6\rangle + |12\rangle + \ldots + |504\rangle + |510\rangle\ \big)\ |1\rangle + \\
&\big(\ |1\rangle + \ |7\rangle + |13\rangle + \ldots + |505\rangle + |511\rangle\ \big)\ |2\rangle + \\
&\big(\ |2\rangle + \ |8\rangle + |14\rangle + \ldots + |506\rangle\ \big)\ \ |4\rangle + \\
&\big(\ |3\rangle + \ |9\rangle + |15\rangle + \ldots + |507\rangle\ \big)\ \ |8\rangle + \\
&\big(\ |4\rangle + |10\rangle + |16\rangle + \ldots + |508\rangle\ \big)\ |16\rangle + \\
&\big(\ |5\rangle + \ |11\rangle + |17\rangle + \ldots + |509\rangle\ \big)\ |11\rangle\ \Big].
\end{aligned} \tag{4.12}$$

Esta característica foi explicitada na equação (4.5). Como a ordem não é uma potência de 2, aqui existe uma pequena diferença: as primeiras duas linhas da equação (4.12) têm 86 termos, enquanto o restante delas tem 85.

Agora é feita uma medida no primeiro registrador[2], gerando um dos seguintes números com igual probabilidade: $\{1, 2, 4, 8, 16, 11\}$. Suponha que o resultado da medida seja 2. Então,

$$|\psi_3\rangle = \frac{1}{\sqrt{86}}\left(|1\rangle + |7\rangle + |13\rangle + \ldots + |505\rangle + |511\rangle\right)|2\rangle\,. \tag{4.13}$$

Observe que o estado $|\psi_3\rangle$ foi renormalizado para ser unitário. Não importa o resultado da medida; o que importa é o padrão periódico de (4.13). O período

[2]Como a medida sempre pode ser feita no final do algoritmo (veja [16], p. 186), este passo não é necessário; serve apenas para simplificar as expressões seguintes.

do estado do primeiro registrador é a solução para o problema, e a transformada de Fourier pode revelar o valor deste período. Então, o próximo passo é aplicar a transformada de Fourier inversa no primeiro registrador de $|\psi_3\rangle$:

$$\begin{aligned} |\psi_4\rangle &= \mathrm{DFT}^\dagger(|\psi_3\rangle) \\ &= \mathrm{DFT}^\dagger\left(\frac{1}{\sqrt{86}}\sum_{a=0}^{85}|6a+1\rangle\right)|2\rangle \\ &= \frac{1}{\sqrt{512}}\sum_{j=0}^{511}\left(\left[\frac{1}{\sqrt{86}}\sum_{a=0}^{85}e^{-2\pi i\frac{6ja}{512}}\right]e^{-2\pi i\frac{j}{512}}|j\rangle\right)|2\rangle\,, \end{aligned} \tag{4.14}$$

onde usamos a equação (4.8) e rearranjamos as somas. A última equação é similar à equação (4.10), mas com uma importante diferença. Na Seção 4.2, assumimos que r divide 2^t. Isto não é verdade nesse caso (6 não divide 512); portanto, não podemos usar a identidade (4.9) para simplificar os termos nos colchetes da equação (4.14). Esses termos nunca se anulam, mas a contribuição principal é ainda em torno de $j = 0,\ 85,\ 171,\ 256,\ 341,\ 427$, que são obtidos de $512k_0/6$ para k_0 de 0 até 5 (compare com a discussão logo após a equação (4.11)). Para nos convencermos, façamos o gráfico da probabilidade de dar o resultado j (no intervalo de 0 até 511), medindo o primeiro registrador do estado $|\psi_4\rangle$. De (4.14), temos que a probabilidade é

$$\mathrm{Prob}(j) = \frac{1}{512\times 86}\left|\sum_{a=0}^{85}e^{-2\pi i\frac{6ja}{512}}\right|^2. \tag{4.15}$$

O gráfico da Prob(j) é mostrado na Figura 4.5. Vemos os picos em torno de $j =$ 0, 85, 171, 256, 341, 427, indicando alta probabilidade de dar um destes valores, ou algum valor muito próximo deles. No intervalo entre eles, a probabilidade é quase zero. A largura dos picos depende de t (número de q-bits no primeiro registrador). O limite inferior de $2^t \geq N^2$ assegura uma alta probabilidade em medir um valor de j carregando a informação desejada. Uma análise cuidadosa da expressão (4.15) é feita em [15], e um estudo meticuloso da forma do pico é feita em [9].

Vamos analisar os possíveis resultados da medida. Se o resultado for $j = 0$ (primeiro pico), o algoritmo não revela o valor de r. Deve ser executado novamente. Escolhamos $x = 2$ e executamos novamente a parte quântica do algoritmo. A probabilidade de dar $j = 0$ é baixa: da equação (4.15) temos que $\mathrm{Prob}(0) = 86/512 \approx 0,167$. Agora suponha que o resultado foi $j = 85$ (ou qualquer valor no segundo pico). Dividimos por 512, resultando 85/512, que é uma aproximação racional de $k_0/6$, para $k_0 = 1$. Como podemos obter r de 85/512?

O método de aproximação por frações contínuas permite-nos extrair a informação desejada. Uma fração contínua de um número racional j_1/j_2 tem a forma

$$\frac{j_1}{j_2} = a_0 + \frac{1}{a_1 + \frac{1}{\ldots + \frac{1}{a_p}}},$$

usualmente representada por $[a_0, a_1, ..., a_p]$, onde a_0 é um inteiro não-negativo e $a_1, ..., a_p$ são positivos. O q-ésimo convergente ($0 \leq q \leq p$) é definido como um

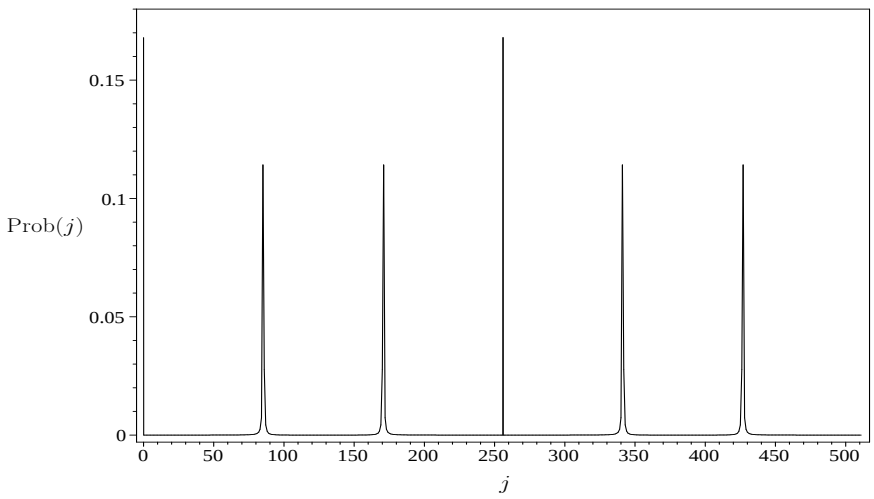

Figura 4.5: Gráfico de Prob(j) em função de j. Compare o formato dos picos deste gráfico com o formato dos picos do gráfico da Figura 4.4.

número racional $[a_0, a_1, ..., a_q]$. Isto é uma aproximação para j_1/j_2 e tem o denominador menor que j_2. Este método é aplicado facilmente pela inversão da fração, seguido pela divisão inteira com resto racional. Invertendo 85/512, temos 512/85, que é igual a $6 + 2/85$. Repetimos o processo com 2/85 até obtermos o numerador 1. O resultado é

$$\frac{85}{512} = \frac{1}{6 + \frac{1}{42 + \frac{1}{2}}}.$$

Assim, os convergentes de 85/512 são 1/6, 42/253 e 85/512. Devemos selecionar os convergentes que tenham um denominador menor que $N = 21$ (já que $r < N$)[3]. Este método fornece 1/6, e então, $r = 6$. Checamos que $2^6 \equiv 1 \bmod 21$, e a parte quântica do algoritmo termina com a resposta correta. A ordem $r = 6$ é um número par, portanto MDC$(2^{(6/2)} \pm 1, 21)$ fornece dois fatores não triviais de 21. Um cálculo direto mostra que qualquer resultado no segundo pico (digamos $81 \leq j \leq 89$) produz o convergente 1/6.

Considere agora o terceiro pico, que corresponde a $k_0 = 2$ da fórmula $k_0/6$. Aplicamos novamente o método de aproximação por frações contínuas, resultando em 1/3, para qualquer j no terceiro pico (digamos $167 \leq j \leq 175$). Neste caso, obtemos um fator de r ($r_1 = 3$), já que $2^3 \equiv 8 \not\equiv 1 \bmod 21$. Rodamos a parte quântica do algoritmo novamente, para achar a ordem de 8. Obteremos $r_2 = 2$. Logo, $r = r_1 r_2 = 3 \times 2 = 6$.

O quarto e quinto picos também fornecem fatores de r. O último pico é similar

[3] A desigualdade $r \leq \varphi(N)$ segue do teorema de Euler: $x^{\varphi(N)} \equiv 1 \mod N$, onde x é um inteiro positivo coprimo com N e φ é a função *totiente* de Euler ($\varphi(N)$ fornece o número de inteiros positivos menores que N, coprimos com N). A desigualdade $\varphi(N) < N$ segue da definição de φ (veja [24], p. 492).

ao segundo, resultando r diretamente.

A avaliação geral da probabilidade de sucesso é a seguinte. A área abaixo de todos os picos é aproximadamente a mesma: $\approx 0,167$. O primeiro e o quarto picos são diferentes dos outros — eles não são espalhados. Para calcular suas contribuições para a probabilidade total, tomamos a base igual a 1. As áreas embaixo do segundo, terceiro, quinto e último picos são calculadas, adicionando a Prob(j), para j rodando em torno do centro de cada pico. Então, em aproximadamente 17% dos casos, o algoritmo falha (1° pico). Em aproximadamente 33% dos casos, o algoritmo retorna r de primeira (2° e 6° picos). Em aproximadamente 50% dos casos, o algoritmo retorna r na segunda rodada ou mais (3°, 4° e 5° picos). Agora, vamos calcular a probabilidade de achar r na segunda rodada. Para o 3° e 5° picos, o fator restante é $r_2 = 2$. O gráfico equivalente para a Figura 4.5, neste caso, tem 2 picos, então o algoritmo retorna r_2 em 50% dos casos. Para o 4° pico, o fator restante é $r = 3$ e o algoritmo retorna r_2 em 66,6% dos casos. Assim, o resultado é $\frac{2\times 50\%+66,6\%}{3}$ de 50%, que é igual a aproximadamente 22%. Resumindo, a probabilidade de sucesso para $x = 2$ é em torno de 55%.

4.5 Transformada de Fourier em termos de Portas Universais

Nas seções precedentes, mostramos que o algoritmo de Shor é um algoritmo probabilístico eficiente, assumindo que a transformada de Fourier poderia ser implementada eficientemente. Nesta seção, decompomos a transformada de Fourier em termos de portas universais: CNOT e portas de 1 q-bit. Esta decomposição permite avaliarmos o custo computacional (complexidade) da transformada discreta de Fourier e mostra como implementá-la em um computador quântico real.

A transformada de Fourier dos estados da base computacional é

$$\mathrm{DFT}(|j\rangle) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N} |k\rangle . \tag{4.16}$$

Notando que o lado direito da equação (4.16) tem N termos e a base computacional tem N estados, temos que o custo computacional do cálculo da transformada de Fourier clássica dos estados da base computacional, usando a equação (4.16), é $O(N^2) = O(2^{2n})$ [4]. Um resultado muito importante em Computação foi o desenvolvimento da transformada de Fourier rápida (FFT), que reduz o custo computacional para $O(n2^n)$ [6]. No nosso contexto, é mais conveniente mostrar esse ganho na complexidade, notando que o lado direito da equação (4.16) é um tipo muito especial de expansão, que pode ser totalmente fatorado. Por exemplo, a transformada de

[4] $O(N^2)$ significa que o custo é proporcional a N^2. Essa notação é útil no cálculo do custo computacional de algoritmos. Para maiores detalhes, veja a referência [17].

Fourier de $\{|0\rangle, |1\rangle, |2\rangle, |3\rangle\}$ pode ser escrita como

$$\begin{aligned}
\mathrm{DFT}(|0\rangle) &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\
\mathrm{DFT}(|1\rangle) &= \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle + i\,|1\rangle}{\sqrt{2}}\right) \\
\mathrm{DFT}(|2\rangle) &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \\
\mathrm{DFT}(|3\rangle) &= \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) \otimes \left(\frac{|0\rangle - i\,|1\rangle}{\sqrt{2}}\right).
\end{aligned} \tag{4.17}$$

Note que no exemplo (4.17), estamos usando a base 2 para fatorar o lado direito da equação (4.16). Agora, vamos fatorar a expressão geral. O primeiro passo é escrever (4.16) na forma

$$\mathrm{DFT}(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} e^{2\pi i j \sum_{l=1}^{n} \frac{k_l}{2^l}} |k_1\rangle \otimes \ldots \otimes |k_n\rangle\,, \tag{4.18}$$

onde o *ket* $|k\rangle$ foi convertido para a base 2, e usamos a expansão $k = \sum_{l=1}^{n} k_l 2^{n-l}$ no expoente. Considerando que a exponencial da soma é o produto das exponenciais, (4.18) transforma-se em um produto (não-comutativo) dos seguintes *kets*:

$$\mathrm{DFT}(|j\rangle) = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} \cdots \sum_{k_n=0}^{1} \prod_{l=1}^{n} \left(e^{2\pi i j \frac{k_l}{2^l}} |k_l\rangle\right). \tag{4.19}$$

Fatorando (4.19), pela troca da soma pelo produto, obtemos:

$$\mathrm{DFT}(|j\rangle) = \frac{1}{\sqrt{2^n}} \prod_{l=1}^{n} \sum_{k_l=0}^{1} \left(e^{2\pi i j \frac{k_l}{2^l}} |k_l\rangle\right). \tag{4.20}$$

Para nos convencermos de que a última equação está correta, façamos o cálculo inverso: simplesmente expandimos o produto na equação (4.20) e, então, colocamos todos os termos da soma no começo da expressão resultante para obter (4.19). Expandindo a soma da equação (4.20) e, então, o produto, obtemos finalmente

$$\begin{aligned}
\mathrm{DFT}(|j\rangle) &= \frac{1}{\sqrt{2^n}} \prod_{l=1}^{n} \left(|0\rangle + e^{2\pi i j/2^l} |1\rangle\right) \\
&= \left(\frac{|0\rangle + e^{2\pi i \frac{j}{2}} |1\rangle}{\sqrt{2}}\right) \otimes \ldots \otimes \left(\frac{|0\rangle + e^{2\pi i \frac{j}{2^n}} |1\rangle}{\sqrt{2}}\right).
\end{aligned} \tag{4.21}$$

O custo do cálculo da equação (4.21) para um $|j\rangle$ é $O(n)$, já que existem n termos no produto. O custo do cálculo da transformada rápida de Fourier clássica de toda a base computacional ainda é exponencial, $O(n2^n)$, já que o cálculo é feito em cada

um dos 2^n elementos da base, um de cada vez. Por outro lado, o computador quântico usa o paralelismo quântico, e a transformada de Fourier do estado

$$|\psi\rangle = \sum_{a=0}^{2^n-1} F(a)\,|a\rangle\,,$$

que tem um número exponencial de termos, é calculada com uma única aplicação da transformada de Fourier quântica. A transformada de Fourier de 2^n elementos da base é calculada simultaneamente. Então, o custo da transformada de Fourier quântica é medida pelo tamanho do circuito. Agora, vamos mostrar que são necessárias $O(n^2)$ portas.

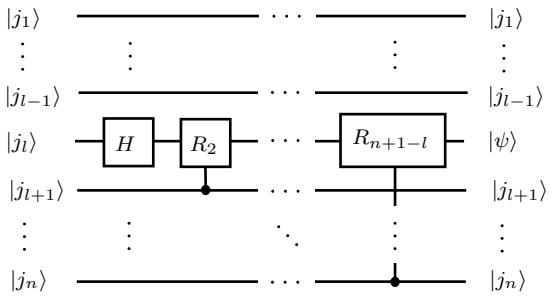

Figura 4.6: Parte do circuito da transformada de Fourier quântica que atua num q-bit $|j_l\rangle$. O valor de todos os q-bits não muda, exceto $|j_l\rangle$, que muda para $|\psi\rangle = \frac{|0\rangle + e^{2\pi i \frac{j}{2^{n+1-l}}}|1\rangle}{\sqrt{2}}$.

Considere o circuito da Figura 4.6. É fácil checar que o valor dos q-bits $|j_m\rangle$, $m \neq l$, não muda. Vamos verificar o caso mais difícil: $|j_l\rangle$. As matrizes unitárias R_k são definidas como

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i \frac{1}{2^k}\right) \end{bmatrix}.$$

Cada porta R_k é controlada pelo q-bit $|j_{k+l-1}\rangle$. Se $j_{k+l-1} = 0$, então R_k deve ser trocado pela matriz identidade (sem ação), e se $j_{k+l-1} = 1$, então R_k é acionada. Isso significa que, para os cálculos propostos, R_k é controlado por $|j_{k+l-1}\rangle$, podendo ser trocado pela seguinte porta de 1 q-bit:

$$CR_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(2\pi i \frac{j_{k+l-1}}{2^k}\right) \end{bmatrix}. \tag{4.22}$$

Para simplificarmos os cálculos, note que

$$H\,|j_l\rangle = \frac{|0\rangle + e^{2\pi i \frac{j_l}{2}}\,|1\rangle}{\sqrt{2}} = CR_1\,|+\rangle\,, \tag{4.23}$$

onde $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$. Então, em vez de usar

$$|\psi\rangle = CR_{n+1-l} \dots CR_2 \, H \, |j_l\rangle \, ,$$

que pode ser lido diretamente da Figura 4.6, usaremos

$$|\psi\rangle = CR_{n+1-l} \dots CR_2 CR_1 \, |+\rangle \, .$$

Definimos

$$PR_{n+1-l} = \prod_{k=n+1-l}^{1} CR_k, \tag{4.24}$$

onde o produto está na ordem reversa. Usando (4.22) e (4.24), obtemos

$$\begin{aligned} PR_{n+1-l} &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & \exp 2\pi i \left(\frac{j_n}{2^{n+1-l}} + \ldots + \frac{j_l}{2} \right) \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 0 & \exp \left(2\pi i \frac{j}{2^{n+1-l}} \right) \end{bmatrix}, \end{aligned} \tag{4.25}$$

onde usamos $j = \sum_{m=1}^{n} j_m 2^{n-m}$ e o fato de que os primeiros $l-1$ termos desta expansão não contribuem — eles são múltiplos inteiros de $2\pi i$ em (4.25). Finalmente, obtemos

$$\begin{aligned} |\psi\rangle &= PR_{n+1-l} \, |+\rangle \\ &= \frac{|0\rangle + e^{2\pi i \frac{j}{2^{n+1-l}}} \, |1\rangle}{\sqrt{2}}. \end{aligned} \tag{4.26}$$

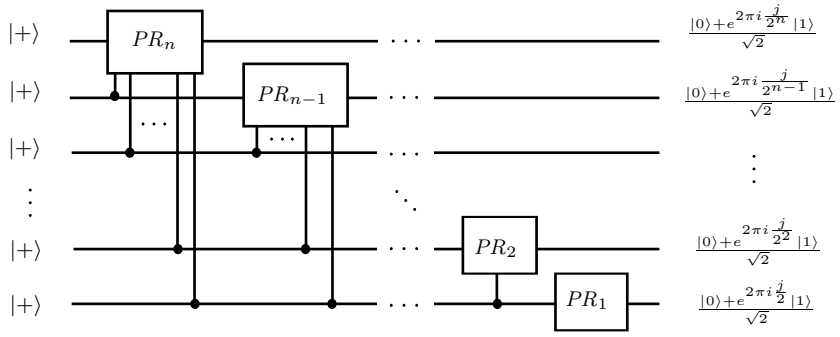

Figura 4.7: Circuito intermediário para a transformada de Fourier quântica. A entrada é tomada como $|+\rangle$, para executar os cálculos propostos, como explicado na equação (4.23). A saída tem a ordem inversa da equação (4.21).

Note que PR_{n+1-l} não pode ser implementado diretamente, atuando apenas no l-ésimo q-bit, porque ele precisa dos valores de j_{l+1} até j_n.

O próximo passo é o circuito da Figura 4.7. Vamos fundir as portas R_k, usando a equação (4.24). As portas PR_k (k de n até 1) são colocadas em seqüência na Figura 4.7. Então, a saída do primeiro q-bit é o último termo da equação (4.21), correspondendo à ação do PR_n no $|\psi_1\rangle$, controlado pelos outros q-bits, que não mudam. O mesmo processo é repetido pelo PR_{n-1} atuando em $|\psi_2\rangle$, gerando o penúltimo termo na equação (4.21), e assim por diante, até reproduzir todos os termos da transformada de Fourier. Agora, resta-nos inverter a ordem dos estados dos q-bits.

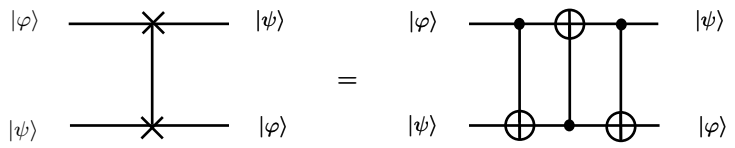

Figura 4.8: Circuito de inversão.

Para inverter os estados de 2 q-bits genéricos, usamos o circuito da Figura 4.8. Vamos mostrar por que o circuito funciona como desejado. Tome a entrada $|\varphi\rangle|\psi\rangle = |0\rangle|1\rangle$. O primeiro CNOT da Figura 4.8 não muda este estado; o CNOT invertido muda para $|1\rangle|1\rangle$; e o último CNOT muda para $|1\rangle|0\rangle$. A saída é $|\psi\rangle|\varphi\rangle$. Se repetirmos o mesmo processo com $|0\rangle|0\rangle$, $|1\rangle|0\rangle$ e $|1\rangle|1\rangle$, concluiremos que o circuito inverte todos os estados da base computacional; portanto, inverte um estado genérico da forma $|\varphi\rangle|\psi\rangle$.

A decomposição não está completa ainda. Resta escrever a porta R_k-controlada em termos de CNOT e portas de 1 q-bit. Esta decomposição é dada na Figura 4.9. A verificação é direta. Basta acompanhar o que acontece na base computacional $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ em ambos os circuitos.

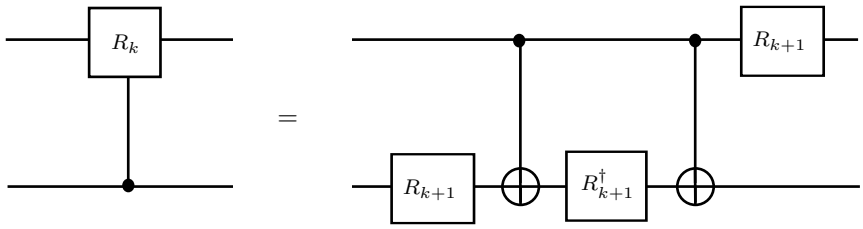

Figura 4.9: Decomposição da porta R_k-controlada em termos de portas universais.

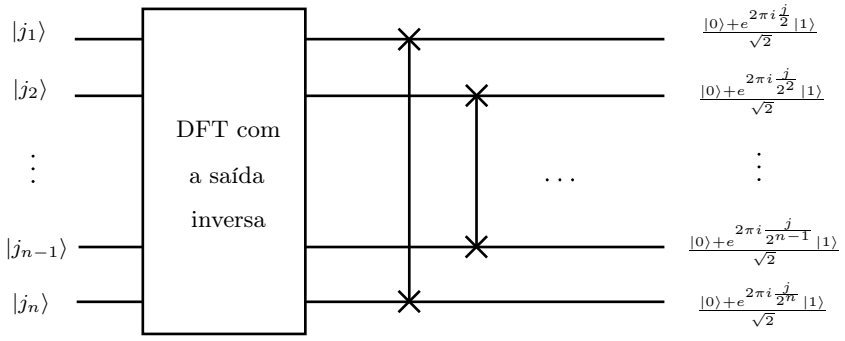

Figura 4.10: Circuito completo para a transformada de Fourier quântica.

O circuito completo para a transformada de Fourier quântica é dado na Figura 4.10. Agora, podemos calcular o custo computacional do circuito quântico da transformada de Fourier. Contando o número de portas elementares nas Figuras de 4.6 até 4.9, obtemos o termo dominante $5n^2/2$, que implica que o custo é $O(n^2)$.

A essa altura, alguém poderia estar se perguntando sobre a decomposição de V_x em termos de portas elementares. V_x é a maior porta da Figura 4.1. Na verdade, Shor declarou no seu artigo [21], em 1997, que V_x é o "gargalo" do algoritmo quântico de fatoração, devido ao tempo e ao espaço consumidos para executar a exponenciação modular (veja [20], p. 10). O gargalo não é tão estreito, já que, usando o método clássico conhecido por "quadrado repetido" e algoritmos de multiplicação de inteiros (veja [24], p. 69), o custo para calcular exponenciação modular é $O(n^3)$. O circuito quântico pode ser obtido do circuito clássico, trocando as portas clássicas irreversíveis pelas equivalentes reversíveis. V_x é um problema em chamadas recursivas do algoritmo, quando x varia. Para cada x, um novo circuito deve ser construído, o que é incômodo no estágio atual do desenvolvimento da computação quântica.

Exercício 4.1 Mostre que a decomposição da Figura 4.9 está correta.

Exercício 4.2 Faça o circuito da transformada de Fourier para o caso de 3 q-bits.

Exercício 4.3 Faça o circuito da transformada de Fourier inversa para o caso de 3 q-bits.

Bibliografia

[1] Aharonov, D. Quantum computation. In *Annual Reviews of Computational Physics*, D. Stauffer, Ed., vol. VI. World Scientific, Jerusalem, 1998, pp. 1–78. (quant-ph/9812037).

[2] Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P. W., Sleator, T., Smolin, J. A., and Weinfurter, H. Elementary gates for quantum computation. *Physical Review A A52*, 5 (1995), 3457–3487. (quant-ph/9503016).

[3] Bennett, C. H., Bernstein, E., Brassard, G., and Vazirani, U. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing 26*, 5 (1997), 1510–1523.

[4] Bernstein, E., and Vazirani, U. Quantum complexity theory. *SIAM Journal on Computing 26*, 5 (1997), 1411–1473.

[5] Boyer, M., Brassard, G., Hoyer, P., and Tapp, A. Tight bounds on quantum searching. *Fortschritte der Physik 46*, 4-5 (1998), 493–506.

[6] Cooley, J., and Tukey, J. An algorithm for machine calculation of complex Fourier series. *Math. Comp. 19* (1965), 297–301.

[7] Deutsch, D. Quantum theory, the church-turing principle and the universal quantum computer. In *Proceedings of the Royal Society of London. Series A* (London, 1985), vol. 400, Royal Society, pp. 97–117.

[8] Deutsch, D., and Jozsa, R. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London. Series A* (London, 1992), vol. 439, Royal Society, pp. 553–558.

[9] Einarsson, G. Probability analysis of a quantum computer. (quant-ph/0303074), unpublished 2003.

[10] Ekert, A., and Jozsa, R. Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physics 68* (1996), 733–753.

[11] Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys. 21* (1982), 467–488.

[12] Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM Symposium on the Theory of Computing* (1996), pp. 212–219. (quant-ph/9605043).

[13] Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letter 79* (1997), 325–328. (quant-ph/9706033).

[14] Hirvensalo, M. *Quantum Computing*. Springer, New York, 2001.

[15] Lomonaco, Jr., S. J. Shor´s quantum factoring algorithm. In *Proceedings of Symposia in Applied Mathematics* (Washington, 2002), S. J. Lomonaco, Jr., Ed., vol. 58, American Mathematical Society, pp. 161–180. (quant-ph/0010034).

[16] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2000.

[17] Papadimitriou, C. H. *Computational Complexity*. Addison Wesley Pub. Co., Massachussetts, 1994.

[18] Pittenger, A. O. *An Introduction to Quantum Computing Algorithms*. Birkhauser, 2000.

[19] Preskill, J. Quantum information and computation. Lecture Notes, California Institute of Technology, unpublished 1998.

[20] Shor, P. W. Algorithms for quantum computation: discrete logarithm and factoring. In *Proc. 35th Annual Symposium on Foundations of Computer Science* (1994), pp. 124–134.

[21] Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing 26*, 5 (1997), 1484 – 1509.

[22] Simon, D. R. On the power of quantum computation. *SIAM Journal on Computing 26*, 5 (1997), 1474–1483.

[23] Vandersypen, L. M., Steffen, M., Breyta, G., Yannoni, C. S., Sherwood, M. H., and Chuang, I. L. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature 414*, 6866 (2001), 883–887.

[24] von zur Gathen, J., and Gerhard, J. *Modern Computer Algebra*. Cambridge University Press, Cambridge, 1999.

Índice