Pontifícia Universidade Católica
do Rio de Janeiro

Roberto Cintra Martins

O Algoritmo de Fatoração de Shor

Dissertação de Mestrado

Dissertação apresentada ao Programa de Pós–graduação em Matemática da PUC–Rio como requisito parcial para obtenção do grau de Mestre em Matemática.

Orientador: Prof. Nicolau Corção Saldanha

Rio de Janeiro
abril 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Roberto Cintra Martins

O Algoritmo de Fatoração de Shor

Dissertação apresentada ao Programa de Pós–graduação em Matemática da PUC–Rio como requisito parcial para obtenção do grau de Mestre em Matemática. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Nicolau Corção Saldanha
Orientador
Departamento de Matemática

Prof. George Svetlichny
Departamento de Matemática - PUC-Rio

Prof. Victor Goulart
Departamento de Matemática - PUC-Rio

Prof. Thiago Barbosa dos Santos Guerreiro
Departamento de Física - PUC-Rio

Prof. Carlos Gustavo Moreira
IMPA

Prof. Ernesto Fagundes Galvão
Departamento de Física - UFF

Prof. Marcio da Silveira Carvalho
Coordenador Setorial do Centro Técnico Científico
Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 05 de abril de 2018

Roberto Cintra Martins

Possui graduação em Engenharia de Eletrônica e Teologia, mestrado em Engenharia de Produção e Teologia, doutorado em Engenharia de Sistemas e Computação.

Ficha Catalográfica

Martins, Roberto Cintra

O Algoritmo de Fatoração de Shor/ Roberto Cintra Martins; orientador: Nicolau Corção Saldanha. — Rio de Janeiro: PUC–Rio, Departamento de Matemática, 2018.

v., 94 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Matemática.

Inclui referências bibliográficas.

1. Matemática – Dissertação 2. Computação quântica; 3. Circuitos quânticos; 4. Fatoração de inteiros; 5. Complexidade computacional I. Saldanha, Nicolau Corção. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. III. Título.

CDD: 510

A João Antônio, Helena e Daniel, arautos do porvir.

Agradecimentos

Agradeço a todos os que me encorajaram a realizar o sonho do reencontro com a paixão intelectual de minha juventude, a Matemática.

Agradeço a minha família de origem, a paulista, que semeou em mim o amor pela busca do conhecimento, e a minha família atual, a carioca, com quem tenho compartilhado as alegrias, desafios e dores de viver em um Rio que já não é mais o mesmo. De modo muito especial agradeço a Aida, mulher de verdade, e a Fernando e a Guilherme, nossos filhos.

O professor George Svetlichny ocupou e ocupa um lugar destacado em toda a trajetória de elaboração deste trabalho pelo acolhimento, dedicação e paciência com os quais soube cuidar deste seu aprendiz — e por ter sugerido um tema absolutamente condizente com minha motivação pela Física Matemática. A ele cabe portanto o agradecimento primeiro no universo desta universidade. Ao professor Nicolau Corção Saldanha, igualmente paciente e dedicado, agradeço pelo tanto que me ensinou, por sua amizade e por ter me proporcionado experiência única e irrepetível, ao menos assim espero, em minha experiência discente.

Contar com a orientação de ambos foi nada menos que um verdadeiro privilégio e quero aqui registrar meu reconhecimento por sua profunda honestidade intelectual, bem como minha gratidão por terem meticulosamente lapidado minha tosca atitude diante da Matemática, buscando sempre refiná-la, malgrado minhas resistências. Com elegância exemplar, cada um a seu modo, ambos souberam tanto incentivar como disciplinar minha aspiração em conduzir o tema em estudo, em alguns momentos de atrevimento, para além da Física Matemática.

Ao professor Thiago Barbosa dos Santos Guerreiro, jovem e destacado cientista que aprendi a respeitar e estimar, agradeço por sua generosa disponibilidade e por sua valiosa contribuição para o avanço de minha compreensão da Mecânica Quântica.

Aos professores Boyan Slavchev Shirakov, Carla Gobel Burlamaqui de Mello, Carlos Tomei e David Francisco Martinez Torres, que me propiciaram o embasamento adequado para a abordagem do tema desta dissertação, quero aqui expressar minha gratidão.

Aos jovens colegas estudantes do Departamento de Matemática da PUC-Rio, que não sabem o quanto me ajudaram a aprender Matemática, partilhando comigo seus talentos e sua amizade, quero que saibam que me sinto gratificado e honrado pela oportunidade de nossa convivência.

Aos funcionários do Departamento de Matemática da PUC-Rio pelo atendimento eficiente, atencioso e impecável a todas as minhas demandas.

Não posso deixar de registrar aqui o apoio que sempre encontrei por parte do Departamento de Engenharia Industrial da PUC-Rio, em especial dos professores Flávia César Teixeira Mendes e Luiz Felipe Roris Rodriguez Scavarda do Carmo, que de forma direta e proativa apoiaram esta incursão de um engenheiro, professor horista daquele departamento, em uma arena próxima à Engenharia — ainda que o seja apenas em aparência, conforme apreendi ao longo de minha trajetória recente como estudante.

Por fim, tendo caminhado nestes últimos três anos como aprendiz no território de fronteira entre a Física e a Matemática, não posso aqui deixar de expressar minha alegria e gratidão face ao próprio mistério maior que até aqui me trouxe e me conduz.

Resumo

Martins, Roberto Cintra; Saldanha, Nicolau Corção. **O Algoritmo de Fatoração de Shor**. Rio de Janeiro, 2018. 94p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

A dissertação apresenta detalhadamente o algoritmo de fatoração de Shor, tanto em termos de sua execução passo a passo como mediante sua representação em forma de circuito, abordando aspectos tanto de sua parte clássica como de sua parte quântica. Inicialmente são apresentados aspectos de teoria dos números indispensáveis para a compreensão do algoritmo e em seguida são desenvolvidos conceitos e propriedades de mecânica quântica e de informação quântica pertinentes. Em atenção ao caráter eminentemente estocástico do algoritmo realiza-se um estudo de sua fonte estocástica e demonstram-se os principais teoremas que embasam a avaliação de sua probabilidade de sucesso. Desenvolvem-se exemplos de simulação clássica do algoritmo. Finalmente, a eficiência do algoritmo de fatoração de Shor é comparada com a de algoritmos clássicos.

Palavras–chave

Computação Quântica. Circuitos Quânticos. Fatoração de inteiros. Complexidade Computacional.

Abstract

Martins, Roberto Cintra; Saldanha, Nicolau Corção (Advisor). **Shor's Factoring Algorithm**. Rio de Janeiro, 2018. 94p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

The dissertation presents in detail Shor's factoring algorithm, including its execution step by step and its representation in the form of a circuit, addressing aspects of both its classical and its quantum parts. Aspects of number theory indispensable to understand the algorithm are presented, followed by a development of concepts and properties of quantum mechanics and quantum information. Considering the eminently stochastic character of the algorithm, a study of its stochastic source is carried out and the main theorems that support the evaluation of its probability of success are proved. Examples of classical simulation of the algorithm are developed. Finally, the efficiency of Shor's factoring algorithm is compared with that of classical algorithms.

Keywords

Quantum Computing; Quantum Circuits; Integer Factoring; Computational Complexity

Sumário

Information is physical.

Rolf Landauer

1
Introdução

"El problema de distinguir números primos de números compuestos y de resolver estos últimos en sus factores primos es conocido como uno de los más importantes y útiles en aritmética ... la dignidad de la ciencia misma parece requerir que todos los medios posibles para la solución de un problema tan elegante y tan célebre sean explorados." C. F. Gauss [1].

1.1
Breve Histórico

Até recentemente a grande maioria dos profissionais de computação considerava e tratava o computador essencialmente como um instrumento da Matemática. Ainda que ninguém pudesse considerá-lo um ente imaterial, muito poucos encaravam o computador como um sistema sujeito às leis naturais que regem o comportamento da matéria, em particular às leis da Física.

Encarar o computador como um sistema dinâmico sujeito a processos físicos, tal como qualquer outro equipamento produzido pelo homem, ou qualquer ente da natureza, era e ainda é uma atitude rara.

As razões pelas quais esse déficit de consciência ou de atitude científica face ao objeto computador surgiu e se estabeleceu desde os primórdios da Ciência da Computação podem vir a ser tema de pesquisa de interesse tanto para historiadores como para futurólogos dessa ciência. Afinal, por que a comunidade científica tomou o computador como mero instrumento da Matemática e não o tratou como um sistema ou processo em si mesmo, submetido às leis naturais?

Não cabe aqui estender tal questionamento mais além: ao contrário, pretendemos aqui apenas registrar o déficit de um olhar de físico dirigido ao computador, que constituiu um fato consumado na comunidade científica até bem pouco tempo.

Neste sentido é notável a contribuição de R. P. Feynman [2], ao apontar para o óbvio que até então não era óbvio para ninguém: assim como qualquer outro ente material, também o computador está sujeito às leis da natureza, em particular às leis da Física, mais em particular àquelas pertinentes à fronteira mais avançada desta ciência, à Mecânica Quântica [1].

Segundo Feynman [2], este fato pode ser usado visando à construção de computadores nos quais se pudessem explorar as implicações das leis quânticas que regem todo o mundo físico.

Em sequência a este insight de Feynman, observa-se o surgimento de um novo campo de pesquisa, a Computação Quântica, com significativa produção acadêmica. Hoje a Computação Quântica começa a superar as fronteiras do ambiente acadêmico e aponta desdobramentos promissores no mundo da produção e comercialização de produtos e serviços.

Contudo a indústria da Computação Quântica é ainda uma entidade remota e a pesquisa acadêmica em torno do tema ainda se refere, com certa cautela, a um "computador quântico hipotético". Entretanto nas sociedades mais avançadas a Mecânica Quântica já se constitui em fundamento científico de importância estratégica para a geração de riqueza no cenário econômico, com implicações em nível global.

Ainda que o sucesso da Mecânica Quântica como a teoria científica mais bem sucedida de todos os tempos possa e deva ser qualificado, não resta dúvida de que com seu advento ocorre uma quebra de paradigma em nosso modo de ver o mundo e em nosso modo de vida. Especificamente no que diz respeito às Ciências da Computação, graças a Feynman hoje reconhecemos conscientemente que mesmo o chamado computador clássico é um produto da Mecânica Quântica entendido, explicado, projetado e construído a partir dela.

Segundo P. Shor [3], Benioff [4], [5], [6] parece ter sido a primeira pessoa a vislumbrar a interação entre computação e Mecânica Quântica, pouco antes do trabalho pioneiro de R. P. Feynman. Embora Benioff não tenha se perguntado se a Mecânica Quântica de fato confere poder extra à computação (esta foi a questão colocada por Feynman), ele mostrou que a operação unitária reversível já era suficiente para desempenhar o poder computacional de uma máquina de Turing, mostrando assim que a Mecânica Quântica aplicada é computacionalmente no mínimo tão poderosa quanto os computadores clássicos. Seu trabalho foi fundamental para tornar possível a pesquisa posterior sobre computadores quânticos.

Ainda segundo P. Shor [3], Feynman [2],[7] parece ter sido o primeiro a sugerir que a Mecânica Quântica pode ser computacionalmente mais pode-

[1] O conteúdo histórico desta seção foi elaborado com base em Shor [3].

rosa do que máquinas de Turing. Ele deu argumentos a respeito de porque a Mecânica Quântica é intrinsecamente cara, do ponto de vista computacional, quando simulada em um computador clássico. Ele também levantou a possibilidade de usar um computador baseado em princípios da Mecânica Quântica para evitar esse problema, implicitamente colocando a questão inversa, qual seja: "Usando a Mecânica Quântica em um computador pode-se calcular mais eficientemente do que em um computador clássico?"

O primeiro a colocar tal pergunta explicitamente foi Deutsch [8]. Para estudar esta questão, ele definiu tanto máquinas quânticas de Turing quanto circuitos quânticos e investigou algumas de suas propriedades.

A mesma questão foi posteriormente abordada por Deutsch e Jozsa [9] e Berthiaume e Brassard [10], [11]. Esses trabalhos mostraram que há problemas que os computadores quânticos podem rapidamente resolver de forma exata, mas que computadores clássicos só podem resolver rapidamente com alta probabilidade e utilizando geradores de números aleatórios. No entanto, esses documentos não mostraram como resolver nenhum problema em tempo polinomial em um computador quântico, que já não fosse anteriormente conhecido e resolvível em tempo polinomial em um computador clássico com a ajuda de um gerador de números aleatórios, permitindo uma pequena probabilidade de erro: esta é a caracterização da classe de complexidade BPP (bound error probability probabilistic polynomial time), a qual é reconhecida como a classe de problemas resolvíveis eficientemente.

Outros trabalhos sobre esse mesmo problema surgiram sob o estímulo da contribuição de Bernstein e Vazirani [12]. Um dos resultados contidos em seu trabalho foi um problema de oráculo (ou seja, um problema envolvendo uma sub-rotina "caixa preta" executável em computador, mas para a qual nenhum código é acessível) que pode ser feito em tempo polinomial em uma máquina de Turing quântica, mas que requer tempo superpolinomial em um computador clássico.

Esse resultado foi melhorado por Simon [13], que deu uma construção muito mais simples de um problema de oráculo que leva tempo polinomial em um computador quântico, mas requer tempo exponencial em um computador clássico. Com efeito, enquanto o problema de Bernstein e Vazirani parece artificial, o de Simon parece bastante natural.

O algoritmo de Simon inspirou o trabalho inovador de Shor, considerado por muitos um autêntico breakthrough na história recente da computação: algoritmos em computação quântica em tempo polinomial para fatoração de grandes inteiros e para logaritmos discretos. De fato, trata-se aqui de dois problemas de teoria dos números que já vinham sendo extensivamente estudados

sem que se tivesse descoberto algoritmo em tempo polinomial em computação clássica (Pomerance [14], Gordon [15], Lenstra e Lenstra [16], Adleman e McCurley [17]). Tais problemas são tão consensualmente considerados difíceis que vários sistemas de criptografia baseados em sua dificuldade foram propostos e aplicados, incluindo o amplamente utilizado criptosistema de chave pública RSA desenvolvido por Rivest, Shamir e Adleman [18].

Mas, nas palavras de Lomonaco [19], "Peter Shor suddenly changed the rules of the game". Shor mostrou que esses problemas podem ser resolvidos em tempo polinomial em um computador quântico com uma pequena probabilidade de erro, o que levou alguns a imaginar que sistemas criptográficos amplamente usados pelo setor financeiro das grandes economias do mundo estivessem com seus dias contados em decorrência desta sua contribuição inovadora.

A construção física de computadores quânticos é em princípio viável de acordo com as leis da Mecânica Quântica. Com efeito, desde a década de 1980 sugestões têm sido feitas visando projetos para construção de tais computadores (Teich, Obermayer e Mahler [20], [21]; Lloyd [22]; Cirac e Zoller [23]; DiVincenzo [24]; Sleator e Weinfurter [25]; Barenco et alli [26]; Chuang e Yamomoto [27]), ainda que dificuldades substanciais na construção de qualquer um deles tenham sido identificadas (Landauer [28]; Unruh [29]; Chuang et alli [27]; Palma, Suominen e Ekert [30]).

Contudo, o estado da arte já indica que os primeiros computadores quânticos já foram fisicamente construídos. Os obstáculos mais difíceis no sentido de sua utilização prática parecem envolver a decoerência de superposições quânticas através da interação do computador com o meio ambiente e a implementação de transformações de estados quânticos com precisão suficiente para dar resultados acurados após muitos passos de cálculo. Ambos esses obstáculos tornam-se ainda mais difíceis à medida em que a dimensão do computador cresce. Por essa razão pode vir a ser possível construir pequenos computadores quânticos, enquanto um aumento de escala em direção à construção de máquinas grandes o suficiente para fazer computações interessantes pode apresentar dificuldades fundamentais.

1.2 Motivação

Não sabemos quão longe nos encontramos do dia em que os sistemas de criptografia que preservam o sigilo bancário em todo o mundo serão superados. Entretanto, é óbvio que a pesquisa em Ciências da Computação não deve se apegar apenas a este fato como único desafio motivador para seu avanço.

Com efeito, ainda que nunca seja construído um computador quântico, a pesquisa em torno à contribuição de Shor ilumina o problema de simulação de Mecânica Quântica em um computador clássico. Qualquer método que faça isso para um operador hamiltoniano arbitrário será necessariamente apto a simular um computador quântico. Portanto, qualquer método geral para simulação de Mecânica Quântica que apresente uma desaceleração polinomial irá levar a um algoritmo em tempo polinomial para fatoração de inteiros.

Nesta dissertação optamos por estudar um algoritmo que é um breakthrough na recente história das Ciências da Computação, mais especificamente no interior da mais jovem de suas áreas, a Computação Quântica.

Esses fatos são razões mais que suficientes para explicar a motivação para a escolha do tema deste trabalho.

1.3 Organização

A dissertação está organizada nos seguintes capítulos:

Capítulo 1: Introdução

Capítulo 2: Aspectos de Teoria dos Números

Capítulo 3: Breve Introdução à Mecânica Quântica Voltada para a Computação Quântica

Capítulo 4: Circuitos Quânticos

Capítulo 5: O Algoritmo de Fatoração de Shor: Apresentação Passo a Passo e Exemplos

Capítulo 6: Demonstração de Teoremas Relativos à Fonte Estocástica do Algoritmo de Shor

Capítulo 7: A Parte Quântica do Algoritmo de Shor como Circuito

Capítulo 8: Considerações Finais

2
Aspectos de Teoria dos Números

Neste capítulo vamos apresentar definições, conceitos, teoremas e propriedades baseados em teoria dos números que serão úteis para o desenvolvimento e o entendimento do tema central deste trabalho, o algoritmo de fatoração de Shor.

Essa apresentação deverá seguir uma sequência que se iniciará por aspectos básicos e elementares, para avançar por meio de um discurso que se pretende breve e coerente em direção a aspectos mais complexos.

2.1
Um Breve Preâmbulo: Notação para Conjuntos Numéricos

Em toda esta dissertação, será adotada a seguinte notação:

$\mathbb{N} = \{0, 1, 2, \ldots\}$ é o conjunto dos números naturais

$\mathbb{N}^* = \{1, 2, \ldots\}$ é o conjunto dos números inteiros positivos

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ é o conjunto dos números inteiros

$\mathbb{Q}$ é conjunto dos números racionais

$\mathbb{R}$ é conjunto dos números reais

$\mathbb{C}$ é conjunto dos números complexos

2.2
Congruência

No algoritmo de Shor o conceito de *ordem multiplicativa* de um inteiro é central. Por sua vez este conceito se baseia em outro, o de *congruência*.

Dizemos que uma relação $\sim$ sobre um conjunto X é de *equivalência* se ela é reflexiva, simétrica e transitiva.

Dado um conjunto X, um $x \in X$ e uma relação de equivalência $\sim$, define-se *classe de equivalência* $\bar{x}$ de x como o conjunto de todos os elementos de X equivalentes a x pela relação $\sim$:

$$\bar{x} = \{y \in X; y \sim x\}$$

O conjunto $\{\bar{x}; x \in X\}$ das classes de equivalência de $\sim$ é chamado *quociente* de X por $\sim$ e é denotado por $X/\sim$.

Sejam $a, b, n \in \mathbb{Z}$, $n \neq 0$. Dizemos que a é *congruente* a b módulo n e escrevemos $a \equiv b \pmod{n}$ se $n|a-b$, ou seja, se a e b deixam o mesmo resto na divisão por n, ou ainda, se $b = a + kn$ com $k \in \mathbb{Z}$.

Dado $n \in \mathbb{Z}$, $n \neq 0$, a relação de congruência módulo n é uma relação de equivalência e portanto define *classes de equivalência*.

O quociente de $\mathbb{Z}$ pela relação de congruência módulo n é chamado *anel de inteiros módulo* n e será denotado por $\mathbb{Z}/(n)$ nesta dissertação.

Dizemos que um inteiro a é *invertível* módulo n se existe $b \in \mathbb{Z}$ com $ab \equiv 1 \pmod{n}$. É fácil provar que tal b existe se e somente se $\text{mdc}(a, n) = 1$.

Define-se o *grupo de unidades* $(\mathbb{Z}/(n))^\times$ como o subconjunto de $\mathbb{Z}/(n)$ formado pelos elementos invertíveis de $\mathbb{Z}/(n)$, isto é, $(\mathbb{Z}/(n))^\times = \{\bar{a} \in \mathbb{Z}/(n); \text{mdc}(a, n) = 1\}$.

2.2.1 Formas de Notação do Resíduo

Dados os inteiros positivos a e n, temos $a \equiv a + kn \pmod{n}$ para todo $k \in \mathbb{Z}$, onde $a + kn$ é o resíduo da congruência, o que nos permite denotá-lo de diferentes formas.

A seguir descreveremos duas notações para o resíduo que serão usadas neste trabalho: o resto da divisão de a por n e o resíduo de menor magnitude.

Notação do Resíduo através do Resto

Dados $a, n \in \mathbb{Z}$, $n \neq 0$, existem $q, r \in \mathbb{Z}$ com $0 \leq r < n$ e $a = nq + r$. Tais q e r são unicamente determinados e são chamados o *quociente* e o *resto* da divisão de a por n. O resto r é por vezes denotado por $a \bmod n$.

Notação através do Resíduo de Menor Magnitude

Dados $a, n \in \mathbb{Z}$, $n > 0$ define-se o resíduo de a módulo n de menor magnitude como o único inteiro $\{a\}_n$ tal que

$$a \equiv \{a\}_n \pmod{n}, \quad -\frac{n}{2} < \{a\}_n \leq \frac{n}{2}$$

É imediato notar que:

$$\{a\}_n = a - n \cdot \text{round}\left(\frac{a}{n}\right) = a - n\left\lfloor \frac{a}{n} + \frac{1}{2} \right\rfloor$$

onde $\lfloor x \rfloor$ denota o único inteiro k tal que $k \leq x < k + 1$.

Concluindo nossas considerações sobre congruência, notamos que na notação do resíduo através do resto $r = a \bmod n$ temos sempre um resíduo r inteiro tal que $0 \leq r \leq n-1$, enquanto na notação através do resíduo de menor magnitude temos sempre que um resíduo $\{a\}_n$ inteiro tal que $-\frac{n}{2} < \{a\}_n \leq \frac{n}{2}$.

2.3 Ordens Aditiva e Multiplicativa

Intimamente ligado ao conceito de congruência encontra-se um teorema básico para o entendimento do algoritmo de fatoração de Shor, o *teorema chinês dos restos*. Entre as diferentes versões de seu enunciado, uma nos é de particular interesse, pois lida com o conceito de *ordem multiplicativa*, que é central no algoritmo.

Por outro lado, nas demonstrações dos teoremas 13 e 14 da seção 2.10 deste capítulo, que fundamentam a avaliação de riscos de fracasso na execução do algoritmo de Shor, faremos uso do conceito de *ordem aditiva*.

Por essas razões importa aqui apresentar os conceitos de ordem aditiva e ordem multiplicativa.

Dados $m, M \in \mathbb{N}^*$ define-se ordem aditiva de m módulo M como o menor inteiro positivo $\text{ord}_M\, m$ tal que

$$m \cdot \text{ord}_M\, m \equiv 0 \pmod{M}$$

Vale observar que a função

$$f : \mathbb{N} \to \mathbb{N}$$

$$x \mapsto m \cdot x \bmod M$$

é periódica com período $\text{ord}_M\, m$, no sentido de que $\text{ord}_M\, m$ é o menor número inteiro positivo tal que $f(x) = f(x + \text{ord}_M\, m)$ para todo $x \in \mathbb{N}^*$.

Analogamente, dados $n, N \in \mathbb{N}^*$ com $\text{mdc}(n, N) = 1$, define-se ordem multiplicativa de n módulo N como o menor inteiro positivo $\text{ord}_N\, n$ tal que

$$n^{\text{ord}_N\, n} \equiv 1 \pmod{N}$$

valendo observação análoga à anterior, de que a função

$$f : \mathbb{N} \to \mathbb{N}$$

$$x \mapsto n^x \bmod N$$

é periódica com período $\mathrm{ord}_N\, n$, no sentido de que $\mathrm{ord}_N\, n$ é o menor número inteiro positivo tal que $f(x) = f(x + \mathrm{ord}_N\, n)$ para todo $x \in \mathbb{N}^*$.

Em particular nas demonstrações dos teoremas 13 e 14 da seção 2.10 deste capítulo vamos aplicar a ordem aditiva $\mathrm{ord}_M\, m$ com $M = 2^f, f \in \mathbb{N}^*$, conforme explicitado na lista a seguir, que contém os elementos a de $\mathbb{Z}/(2^f) = \{0, 1, \ldots, 2^{f-1}\}$, segundo sua ordem nesse anel de inteiros.

Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 1$ tem-se um único elemento, $a = 0$.
Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 2$ tem-se um único elemento, $a = 2^{f-1}$.
Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 4$ tem-se dois elementos, $a = 2^{f-2}$ e $a = 3 \cdot 2^{f-2}$.
Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 8$ tem-se quatro elementos, $a = 2^{f-3}, 3 \cdot 2^{f-3}, 5 \cdot 2^{f-3}, 7 \cdot 2^{f-3}$.

...

Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 2^{f-1}$ tem-se 2^{f-2} elementos, $a = 2, 3 \cdot 2, 5 \cdot 2, \ldots, 2^f - 2$.
Com $\mathrm{ord}_{\mathbb{Z}/(2^f)}\, a = 2^f$ tem-se 2^{f-1} elementos, $a = 1, 3, 5, \ldots, 2^f - 1$.

Neste capítulo voltaremos a abordar o conceito de ordem multiplicativa de forma mais elaborada na seção 2.8, relacionando-a a anéis de inteiros módulo n, a grupos de unidade, à função totiente de Euler e a raízes primitivas.

2.4 Teorema Chinês dos Restos ou Teorema dos Restos Chinês

Na literatura encontram-se diversas versões para o enunciado deste importante teorema, das quais mencionamos a seguir apenas duas.

Uma versão mais fraca do enunciado é muito frequentemente encontrada na literatura. Esta versão não remete diretamente ao conceito de *ordem multiplicativa*:

Teorema 1 (Teorema dos restos chinês - enunciado mais fraco)**.** *Seja dado o sistema de congruências:*

$x \equiv b_1 \pmod{a_1}$,
$x \equiv b_2 \pmod{a_2}$,
...
$x \equiv b_k \pmod{a_k}$

onde a_i, a_j são primos entre si para $i \neq j$.

Então há uma solução x que satisfaz todas as congruências. Além disso quaisquer duas soluções são congruentes entre si módulo $a_1 a_2 \cdots a_k$.

A seguir apresentamos um enunciado mais forte que é de maior interesse para nosso trabalho, pois está diretamente relacionado à *ordem multiplicativa* e será utilizado na demonstração dos teoremas 14 e 15 ao final deste capítulo. Estes dois teoremas são centrais para a avaliação da probabilidade de sucesso na execução do algoritmo de fatoração de Shor.

Teorema 2 (Teorema dos restos chinês - enunciado mais forte)**.** *Sejam $a_1, \ldots, a_k$ inteiros positivos primos entre si (dois a dois). Seja $A = a_1 a_2 \cdots a_k$. Então o mapa*

$$\begin{aligned} f : \mathbb{Z}/(A) &\to \mathbb{Z}/(a_1) \oplus \mathbb{Z}/(a_2) \oplus \cdots \oplus \mathbb{Z}/(a_k) \\ b \bmod A &\mapsto (b \bmod a_1, b \bmod a_1, \ldots, b \bmod a_k) \end{aligned}$$

é um isomorfismo no sentido de ser uma bijeção que preserva somas e produtos, explícito e natural.

Corolário 1. *Sejam $a_1, \ldots, a_k$ inteiros positivos primos entre si (dois a dois). Seja $A = a_1 a_2 \cdots a_k$. A restrição*

$$f|_{(\mathbb{Z}/(A))^\times} : (\mathbb{Z}/(A))^\times \to (\mathbb{Z}/(a_1))^\times \oplus (\mathbb{Z}/(a_2))^\times \oplus \cdots \oplus (\mathbb{Z}/(a_k))^\times$$

é um isomorfismo, no sentido de ser uma bijeção que respeita produtos, explícito mas não natural.

Corolário 2. *Em particular, para N_1 e N_2 primos entre si temos*

$$\operatorname{ord}_{N_1 N_2} m = \operatorname{mmc}(\operatorname{ord}_{N_1} m, \operatorname{ord}_{N_2} m).$$

2.5 Frações Contínuas

2.5.1 Definição e Procedimento de Obtenção

No algoritmo de Shor a **ordem multiplicativa** $ord_N m$, onde m é escolhido aleatoriamente e N é o inteiro a ser fatorado, é obtida mediante a notação de um racional como fração contínua.

Seja c um número real. Sabe-se que c pode ser escrito como *fração contínua* na forma

$$c = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots}}}$$

onde a_0 é um natural e $a_1, a_2, \ldots$ são inteiros positivos.

Os valores de $a_0, a_1, \ldots$ são obtidos recursivamente pelo procedimento:
Faça $s_0 = c$ e tome $a_0 = \lfloor s_0 \rfloor, r_1 = s_0 - a_0$
Se $r_1 = 0$, então $c = s_0 = a_0$, FIM
Se $r_1 \neq 0$, faça $s_1 = \frac{1}{r_1}, a_1 = \lfloor s_1 \rfloor, r_2 = s_1 - a_1$
...
Dado s_k, tome $a_k = \lfloor s_k \rfloor, r_{k+1} = s_k - a_k$
Se $r_{k+1} = 0$, faça $M = k, a_M = a_k$, FIM
Se $r_{k+1} \neq 0$, faça $s_{k+1} = \frac{1}{r_{k+1}}, a_{k+1} = \lfloor s_{k+1} \rfloor, r_{k+2} = s_{k+1} - a_{k+1}$
...

A execução passo a passo desse procedimento leva às seguintes igualdades, que resultam na notação do real c como fração contínua:

$$c = s_0 = a_0 + r_1 = a_0 + \frac{1}{s_1} = a_0 + \frac{1}{a_1 + r_2} =$$

$$= a_0 + \frac{1}{a_1 + \frac{1}{s_2}} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + r_3}} = \cdots = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots}}}$$

No caso de c ser um racional, o procedimento recursivo acima termina em um número finito de passos e teremos o inteiro positivo k tal que $r_{k+1} = 0$.

Fazemos então $M = k, a_M = a_k$ e a notação de c como fração contínua toma a forma

$$c = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{... + \cfrac{1}{a_M}}}}$$

2.5.2 Convergentes

Define-se o *n-ésimo convergente* de c como

$$c_n = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots + \cfrac{1}{a_n}}}}$$

No caso de c ser um racional, temos obviamente $0 \leq n \leq M$.
Seja $\frac{p_n}{q_n}$ a forma irredutível do n-ésimo convergente c_n.
Os valores p_n e q_n são obtidos recursivamente pelas relações:

$$p_0 = a_0, \qquad p_1 = a_1 a_0 + 1, \qquad p_n = a_n p_{n-1} + p_{n-2}$$

$$q_0 = 1, \qquad q_1 = a_1, \qquad q_n = a_n q_{n-1} + q_{n-2}.$$

2.5.3 Frações Contínuas e Algoritmo de Euclides

Dado o racional c, seja $\frac{p}{q}$ sua forma irredutível. A relação entre fração contínua e o algoritmo de Euclides fica clara quando aplicamos tal algoritmo para obter $d = \text{mdc}(p, q)$ quando $\frac{p}{q}$ é a forma irredutível de c.

Temos então, seguindo passo a passo o algoritmo de Euclides:
$p = a_0 q + r_1, 0 \leq r_1 < q$. Se $r_1 = 0, d = q = 1$, FIM.
Se $r_1 \neq 0, q = a_1 r_1 + r_2, 0 \leq r_2 < r_1$. Se $r_2 = 0, d = r_1 = 1$, FIM.
Se $r_2 \neq 0, r_1 = a_2 r_2 + r_3, 0 \leq r_3 < r_2$. Se $r_3 = 0, d = r_2 = 1$, FIM.

...

Se $r_k \neq 0, r_{k-1} = a_k r_k + r_{k+1}, 0 \leq r_{k+1} < r_k$. Se $r_{k+1} = 0, d = r_k = 1$, FIM.

...

Terminando com

$$d = \text{mdc}(p, q) = 1$$

Em sua aparente ingenuidade, a execução passo a passo do algoritmo de Euclides para obter o máximo divisor comum dos inteiros p e q primos entre si parece algo sem sentido. Entretanto ela auxilia o entendimento e a visualização dos sucessivos denominadores da notação como fração contínua do racional $c = \frac{p}{q}$ e sugere um procedimento eficiente para a obtenção dessa notação, pois temos:

$p = a_0 q + r_1, 0 \leq r_1 < q \rightarrow \frac{p}{q} = a_0 + \frac{1}{\frac{q}{r_1}}$,
$q = a_1 r_1 + r_2, 0 \leq r_2 < r_1 \rightarrow \frac{q}{r_1} = a_1 + \frac{1}{\frac{r_1}{r_2}}$,
$r_1 = a_2 r_2 + r_3, 0 \leq r_3 < r_2 \rightarrow \frac{r_1}{r_2} = a_2 + \frac{1}{\frac{r_2}{r_3}}$,

...

$r_{M-2} = a_{M-1} r_{M-1} + r_M, 0 \leq r_M < r_{M-1} \rightarrow \frac{r_{M-2}}{r_{M-1}} = a_{M-1} + \frac{1}{\frac{r_{M-1}}{r_M}}$,
$r_{M-1} = a_M r_M + r_{M+1}$ com $r_{M+1} = 0 \rightarrow d = r_M = 1 \rightarrow a_M = r_{M-1} =$ antepenúltimo resto na execução passo a passo do algoritmo de Euclides.

...

Em particular, no algoritmo de Shor a ordem multiplicativa de m módulo N será obtida como o denominador q_n do convergente $c_n = \frac{p_n}{q_n}$ do racional c. Por conta disso o teorema a seguir é de interesse, pois nos oferece uma condição suficiente para que um racional seja um convergente da expansão por frações contínuas de um real.

Teorema 3. *Se x é um número real e a e b são inteiros com $b > 0$ satisfazendo $|\frac{a}{b} - x| < \frac{1}{2b^2}$ então $\frac{a}{b}$ é um convergente da fração contínua de x.*

2.6 Exponenciação Modular

Intrínseca à definição de *ordem multiplicativa* de um inteiro m módulo N encontra-se a necessidade de operar o cálculo de potências de m em aritmética modular.

Dados dois inteiros positivos a e m com $m > 1$, pode-se obter $m^a \bmod N$ aplicando-se o algoritmo clássico de exponenciação modular como a seguir.

Seja $a = a_0 + 2a_1 + 2^2a_2 + ... + 2^s a_s$ a representação binária de a.

onde $a_i \in \{0, 1\}, i = 0, 1, ..., s-1$ e $a_s = 1$.

Inicie com $m_0 = m$ e então para $i = 1, ..., s$ calcule

$$m_i = m_{i-1}^2 m^{a_{s-i}} \bmod N$$

Terminando com

$$m_s = m^a \bmod N$$

pois é claro que

$$m_s = m^{a_0+2a_1+2^2a_2+...+2^s a_s} \bmod N$$

2.7
Números Primos e a Função Totiente de Euler

2.7.1
A Função Totiente de Euler

A função totiente de Euler está presente no enunciado de teoremas que asseguram limitantes inferiores positivos para a probabilidade de sucesso na obtenção da ordem multiplicativa $\mathrm{ord}_N\, m$ — um passo decisivo no algoritmo de fatoração de Shor.

A função totiente de Euler associa a cada inteiro positivo n o número de inteiros positivos menores que n e primos com n:

$$\phi : \mathbb{N}^* \to \mathbb{N}^*$$

$$n \mapsto |(\mathbb{Z}/(n))^\times| = |\{a \in \mathbb{Z}; 1 \leq a \leq n-1, \mathrm{mdc}(a, n) = 1\}|$$

2.7.2
Alguns Teoremas Fundamentais Envolvendo Números Primos e a Função Totiente de Euler

A seguir apresentamos alguns teoremas envolvendo números primos e a função totiente de Euler, pertinentes à teoria dos números elementar, que se aplicam ao tema desta dissertação. Suas demonstrações são bastante conhecidas e encontram-se disponíveis na literatura.

Teorema 4 (Teorema fundamental da aritmética)**.** *Seja $n \in \mathbb{N}, n \geq 2$. Pode-se escrever n de uma única forma, a menos da ordem dos fatores, como um produto $n = p_1 p_2 \cdots p_m$, onde $m \in \mathbb{N}^*$ e $p_1, p_2, \ldots p_k$ são primos.*

Outra forma de escrever a fatoração acima é $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, onde $p_1, p_2, \ldots p_k$ são primos distintos e $e_1, e_2, \ldots e_k \in \mathbb{N}^*$.

Teorema 5. *Se p é primo e $n \in \mathbb{N}^*$ então $\phi(p^n) = p^n - p^{n-1}$*

Teorema 6. *Se m e n são primos entre si então $\phi(mn) = \phi(m)\phi(n)$*

Teorema 7. *Se $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ onde $p_1, p_2, \ldots p_k$ são primos distintos e $e_1, e_2, \ldots e_k \in \mathbb{N}$ então*

$$\phi(n) = n(1 - \frac{1}{p_1})(1 - \frac{1}{p_2}) \cdots (1 - \frac{1}{p_k})$$

Corolário 3. *Se $n = p_1 p_2 \ldots p_k$ com $p_1, p_2, \ldots p_k$ primos distintos então*

$$\phi(n) = (p_1 - 1)(p_2 - 1) \cdots (p_k - 1)$$

Teorema 8 (Teorema de Euler - Fermat)**.** *Se* $\mathrm{mdc}(a, m) = 1$ *então* $a^{\phi(m)} \equiv 1 \pmod{m}$.

Um caso particular desse teorema é o

Teorema 9 (Pequeno Teorema de Fermat)**.** *Seja p um número primo. Qualquer inteiro a satisfaz $a^p \equiv a \pmod{p}$ e todo inteiro a não divisível por p satisfaz $a^{p-1} \equiv 1 \pmod{p}$.*

2.8
Ordem Multiplicativa e Raízes Primitivas

Dada a classe de equivalência $\bar{m} \in (\mathbb{Z}/(N))^{\times}$ define-se *ordem multiplicativa* de $\bar{m}$ como o menor inteiro positivo $\mathrm{ord}_N\, \bar{m}$ tal que $\bar{m}^{\mathrm{ord}_N\, \bar{m}} = \bar{1}$ em $\mathbb{Z}/(N)$.

Se $m, N \in \mathbb{Z}$ com $\mathrm{mdc}(m, N) = 1$, define-se *ordem multiplicativa* de m módulo N, denotada por $\mathrm{ord}_N\, m$, como a ordem de $\bar{m} \in (\mathbb{Z}/(N))^{\times}$.

Notamos que pelo teorema de Euler-Fermat tem-se $\mathrm{ord}_N\, m | \phi(N)$. Se $\mathrm{ord}_N\, a = \phi(N)$, dizemos que a é *raiz primitiva módulo N*.

Associado ao conceito de raiz primitiva temos o seguinte teorema, que será importante nas demonstrações dos teoremas 13 e 14 apresentados na seção 2.10 deste capítulo.

Teorema 10. *Existe alguma raiz primitiva módulo N se, e somente se,* $N = 2, N = 4, N = p^k$ *ou* $N = 2p^k$*, onde p é primo ímpar.*

Este teorema encontra-se demonstrado em [31].

Em particular, nas demonstrações dos teoremas 13 e 14 usaremos a raiz primitiva de p^n com p primo ímpar como base para transformações logarítmicas.

O resultado básico mais importante sobre ordem multiplicativa é que

$$m^t \equiv 1 \pmod{N} \Longleftrightarrow \mathrm{ord}_N\, m | t$$

Lembramos que $\mathrm{ord}_N\, m$ é o período da função

$$f : \mathbb{N} \to \mathbb{N}$$

$$x \mapsto m^x \bmod N$$

e que

$$\mathrm{ord}_{N_1 N_2}\, m = \mathrm{mmc}(\mathrm{ord}_{N_1}\, m, \mathrm{ord}_{N_2}\, m)$$

2.9
Teoremas mais Avançados Envolvendo a Função Totiente de Euler

Conforme já mencionamos, a função totiente de Euler é utilizada na demonstração de teoremas que asseguram limitantes inferiores positivos para as probabilidades de sucesso de certos passos do algoritmo de fatoração de Shor. Na demonstração desses teoremas comparecem os teoremas e colorários a seguir, que vão além do campo da teoria dos números elementar. Suas demonstrações exigiriam a elaboração de temas específicos da teoria dos

números e fogem ao escopo desta dissertação.

Teorema 11. *Existe $C > 0$ tal que para N inteiro positivo suficientemente grande tem-se* $\frac{\phi(N)}{N} \geq \frac{C}{\log_2 \log_2 N}$

O uso de base 2 explícito no enunciado não é indispensável para que se garantam limitantes inferiores positivos para as probabilidades de sucesso mencionadas. Contudo na demonstração de teoremas referentes a essas probabilidades $\log_2$ é explicitado.

Este teorema encontra-se demonstrado em [32].

Teorema 12. *Seja N inteiro positivo. Então* $\liminf \frac{\phi(N)}{\frac{N}{\ln \ln N}} = e^{-\gamma}$ *onde γ denota a constante de Euler*

$$\gamma = 0.57721566490153286061..., \; e^{-\gamma} = 0.5614594836...$$

Este teorema encontra-se demonstrado em [33].

Corolário 4. *Seja P inteiro positivo. Então*

$$\frac{\phi(P)}{\frac{P}{\ln \ln P}} \geq e^{-\gamma} - \epsilon(P)$$

onde $\epsilon(P)$ é uma sequência monótona descrescente de números reais positivos que converge para 0.

Corolário 5. *Sejam N e m inteiros positivos e seja $P = ord_N m$. Então*

$$\frac{\phi(P)}{P} \geq \frac{e^{-\gamma} - \epsilon(P)}{\ln \ln P} \geq \frac{e^{-\gamma} - \epsilon(P)}{\ln \ln N} = \frac{e^{-\gamma} - \epsilon(P)}{\ln \ln 2 + \ln \log_2 N} \geq \frac{e^{-\gamma} - \epsilon(P)}{\ln 2} \frac{1}{\log_2 \log_2 N}$$

Aqui também o uso de base 2 explícito no enunciado não é indispensável, mas conveniente tendo em vista a demonstração de teoremas referentes à probabilidade de sucesso do algoritmo de Shor.

2.10 Probabilidades

Ao contrário dos teoremas das seções 2.7 e 2.9, os três teoremas a seguir e seus corolários se inserem de forma significativa e direta no escopo desta dissertação, pois enunciam explicitamente limitantes superiores ou inferiores para as probabilidades de ocorrência de fracasso ou de sucesso, respectivamente, no algoritmo de fatoração de Shor. Por essa razão vamos aqui proceder às suas respectivas demonstrações.

Teorema 13. *Seja dado $N \in \mathbb{N}$, N ímpar. Escolhe-se um inteiro m uniformemente distribuído em $(\mathbb{Z}(N))^\times$. Então Prob(*$\mathrm{ord}_N\, m$ *ser ímpar)* $\leq \frac{1}{2^k}$, *onde k é o número de fatores primos distintos de N. Além disso, tem-se que Prob(*$\mathrm{ord}_N\, m$ *ser ímpar) é uma potência inteira positiva de* $\frac{1}{2}$.

Demonstração. Pelo teorema fundamental da aritmética temos

$$N = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$$

onde $p_1, p_2, ...p_k$ são primos ímpares distintos e $e_1, e_2, ...e_k \in \mathbb{N}^*$.

Sabemos do teorema 5 que para p primo e $n \in \mathbb{N}^*$ temos $\phi(p^n) = p^{n-1}(p-1)$ onde ϕ é a função totiente de Euler. Logo nestas condições $\phi(p^n)$ é par.

Decorre do corolário 1 que

$$(\mathbb{Z}/(N))^\times = (\mathbb{Z}/(p_1^{e_1}))^\times \oplus \cdots \oplus (\mathbb{Z}/(p_k^{e_k}))^\times$$

Aplicando transformações logarítmicas, com bases dadas pelas respectivas raízes primitivas, temos

$$(\mathbb{Z}/(N))^\times = \mathbb{Z}/(\phi(p_1^{e_1})) \oplus \cdots \oplus \mathbb{Z}/(\phi(p_k^{e_k}))$$

Do teorema do resto chinês segue

$$(\mathbb{Z}/(N))^\times = \mathbb{Z}/(p_1 - 1) \oplus \mathbb{Z}/(p_1^{e_1 - 1}) \cdots \oplus \mathbb{Z}/(p_k - 1) \oplus \mathbb{Z}/(p_k^{e_k - 1})$$

onde $p_i - 1$ é par e pode ser fatorado como $2^{f_i} q_{i1}^{g_{i1}} \cdots q_{il}^{g_{il}}$ com $q_{i1}, \cdots q_{il}$ primos ímpares, $f_i > 0, l \geq 0$, $p_i \equiv 2^{f_i} + 1 \pmod{2^{f_i+1}}$ e teremos:

$$(p_1 - 1) \cdots (p_k - 1) = 2^{f_1} q_{11}^{g_{11}} \cdots q_{1l}^{g_{1l}} \cdots 2^{f_k} q_{k1}^{g_{k1}} \cdots q_{kl}^{g_{kl}}$$

Aplicando tal fatoração à expressão anterior temos

$$(\mathbb{Z}/(N))^{\times} = (\mathbb{Z}/(p_1^{e_1}))^{\times} \oplus \cdots \oplus (\mathbb{Z}/(p_k^{e_k}))^{\times} = G_0 \oplus G_1$$

onde

$$G_0 = \mathbb{Z}/(2^{f_1}) \oplus \cdots \oplus \mathbb{Z}/(2^{f_k}),$$

cuja ordem é $2^{(f_1+\cdots+f_k)}$, e

$$G_1 = \mathbb{Z}/(q_{11}^{g_{11}})...\oplus\mathbb{Z}/(q_{1l}^{g_{1l}})\oplus\mathbb{Z}/(p_1^{e_1-1})\oplus...\oplus\mathbb{Z}/(q_{k1}^{g_{k1}})\cdots\oplus\mathbb{Z}/(q_{kl}^{g_{kl}})\oplus\mathbb{Z}/(p_k^{e_k-1}),$$

que tem ordem ímpar.

Devemos sortear aleatoriamente um inteiro m uniformemente distribuído em $(\mathbb{Z}/(N))^{\times}$, com interesse voltado para a probabilidade de que a ordem multiplicativa de m módulo N seja ímpar. Notamos que esse procedimento é equivalente ao sorteio aleatório de um elemento a em $G_0 \oplus G_1$, com interesse voltado para a probabilidade de que a ordem aditiva de a módulo N seja ímpar. Com base nessa equivalência doravante passaremos a utilizar exclusivamente o segundo procedimento, isto é, adotaremos sempre $G_0 \oplus G_1$, e não mais $(\mathbb{Z}(N))^{\times}$, como espaço amostral para nosso sorteio aleatório.

Para $a = (a_0, a_1) \in G_0 \oplus G_1$, temos pelo corolário 2

$$\mathrm{ord}_{G_0 \oplus G_1}(a_0, a_1) = \mathrm{mmc}(\mathrm{ord}_{G_0} a_0, \mathrm{ord}_{G_1} a_1)$$

Por outro lado, para $a \in G_0 = \mathbb{Z}/(2^{f_1}) \oplus \cdots \oplus \mathbb{Z}/(2^{f_k}), a = (a_1 \ldots a_k)$, temos pelo corolário 2

$$\mathrm{ord}_{G_0} a = \mathrm{mmc}(\mathrm{ord}_{\mathbb{Z}/(2^{f_1})} a_1, \ldots, \mathrm{ord}_{\mathbb{Z}/(2^{f_k})} a_k)$$

Ao sortearmos um elemento $a = (a_0, a_1)$ uniformemente distribuído em $G_0 \oplus G_1$ a paridade da ordem de a módulo N dependerá *somente* da escolha aleatória de um elemento a_0 uniformemente distribuído em G_0, pois G_1 tem ordem ímpar.

Ora, $\mathbb{Z}/(2^{f_i}) = \{0, 1, ..., 2^{f_i} - 1\}$ possui 2^{f_i} elementos, dos quais *somente um elemento* é de ordem ímpar módulo N: o elemento 0, que tem ordem 1 módulo N. Os demais elementos de $\mathbb{Z}/(2^{f_i})$ têm ordem dada por 2^n, com $n = 1, ..., f_i$. Portanto em G_0 temos *somente um elemento* de ordem ímpar módulo N: o elemento $(0, 0, \ldots, 0)$, que tem ordem 1 módulo N.

Notando que G_0 possui $2^{f_1+\ldots+f_k}$ elementos, temos Prob($ord_N a$ ser ímpar)$=\frac{1}{2^{f_1+\cdots+f_k}}$

Como $f_i \geq 1$ para $i \in \{1, 2, \ldots, k\}$, o teorema está demonstrado. □

Corolário 6. *Seja dado $N \in \mathbb{N}$, N ímpar, $N \geq 3$. Escolhe-se um inteiro m uniformemente distribuído em $(\mathbb{Z}(N))^\times$. Então Prob(*$\mathrm{ord}_N\, m$ *ser ímpar)* $\leq \frac{1}{2}$. *Para N composto com pelo menos dois fatores primos distintos tem-se Prob(*$\mathrm{ord}_N\, m$ *ser ímpar)* $\leq \frac{1}{4}$.

Teorema 14. *Seja $N \in \mathbb{N}$ ímpar com k fatores primos distintos, $k \geq 2$. Sorteia-se aleatoriamente $m \in (\mathbb{Z}/(N))^\times$, com distribuição uniforme de probabilidades. Então*

$$\mathrm{Prob}((\mathrm{ord}_N\, m \text{ ser par}) \wedge (m^{\frac{\mathrm{ord}_N\, m}{2}} + 1 \equiv 0 \pmod N)) < \frac{1}{2^k - 1}$$

Observação: O caso $k = 0$ é desinteressante. Para o caso $k = 1$ a desigualdade acima é trivialmente satisfeita.

Demonstração. Consideremos inicialmente um grupo aditivo cíclico com c elementos

$$G = \mathbb{Z}/(c) = \{0, 1, \ldots, c-1\}$$

e observemos que G tem um único elemento de ordem 2 se c é par, a saber $\frac{c}{2}$, e nenhum elemento de ordem 2 se c é ímpar.

Seja $N = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$ onde $p_1, p_2, ...p_k$ são primos ímpares distintos e $e_1, e_2, ...e_k \in \mathbb{N}^*$ a decomposição (única) de N em fatores primos.

Como já visto na demonstração do teorema 13, decorre do corolário 1 que

$$(\mathbb{Z}/(N))^\times = (\mathbb{Z}/(p_1^{e_1}))^\times \oplus \cdots \oplus (\mathbb{Z}/(p_k^{e_k}))^\times$$

Aplicando transformações logarítmicas, com bases dadas pelas respectivas raízes primitivas, temos

$$(\mathbb{Z}/(N))^\times = \mathbb{Z}/(\phi(p_1^{e_1})) \oplus \cdots \oplus \mathbb{Z}/(\phi(p_k^{e_k}))$$

Do teorema do resto chinês segue

$$(\mathbb{Z}/(N))^\times = \mathbb{Z}/(p_1 - 1) \oplus \mathbb{Z}/(p_1^{e_1 - 1}) \cdots \oplus \mathbb{Z}/(p_k - 1) \oplus \mathbb{Z}/(p_k^{e_k - 1})$$

Em particular estamos interessados em $\mathrm{Prob}(m^{\frac{\mathrm{ord}_N\, m}{2}} \equiv -1 \pmod N)$. Consideremos o mapa

$$\begin{aligned} (\mathbb{Z}(N))^\times &\to (\mathbb{Z}/(p_1^{e_1}))^\times \oplus \cdots \oplus (\mathbb{Z}/(p_k^{e_k}))^\times \\ a \bmod N &\mapsto (a \bmod p_1^{e_1}, \ldots, a \bmod p_k^{e_k}), \end{aligned}$$

Nesse mapa notamos em particular que

$$-1 \mapsto (-1, \ldots, -1)$$

Já no mapa

$$(\mathbb{Z}/(p_1^{e_1}))^\times \oplus \cdots \oplus (\mathbb{Z}/(p_k^{e_k}))^\times \mapsto \mathbb{Z}/(\phi(p_1^{e_1})) \oplus \cdots \oplus \mathbb{Z}/(\phi(p_k^{e_k}))$$

notamos que

$$(-1, \ldots, -1) \mapsto (\frac{\phi(p_1^{e_1})}{2}, \ldots, \frac{\phi(p_k^{e_k})}{2})$$

Finalmente no mapa

$$\mathbb{Z}/(\phi(p_1^{e_1})) \oplus \cdots \oplus \mathbb{Z}/(\phi(p_k^{e_k})) \rightarrow$$
$$\rightarrow \mathbb{Z}/(p_1 - 1) \oplus \mathbb{Z}/(p_1^{e_1 - 1}) \oplus \cdots \oplus \mathbb{Z}/(p_k - 1) \oplus \mathbb{Z}/(p_k^{e_k - 1})$$

notamos que

$$(\frac{\phi(p_1^{e_1})}{2}, \ldots, \frac{\phi(p_k^{e_k})}{2}) \mapsto (\frac{p_1 - 1}{2}, 0, \ldots, \frac{p_k - 1}{2}, 0)$$

Em $\mathbb{Z}/(p_i - 1)$ temos $p_i - 1$ par, podendo ser fatorado como $p_i - 1 = 2^{f_i} q_{i1}^{g_{i1}} \cdots q_{il}^{g_{il}}$ com $f_i > 0, l \geq 0$.

Temos então o mapa

$$\mathbb{Z}/(p_1 - 1) \oplus \mathbb{Z}/(p_1^{e_1 - 1}) \oplus \cdots \oplus \mathbb{Z}/(p_k - 1)) \oplus \mathbb{Z}/(p_k^{e_k - 1}) \rightarrow$$
$$\rightarrow \mathbb{Z}/(2^{f_1}) \oplus \mathbb{Z}/(q_{11}^{g_{11}}) \oplus \cdots \oplus \mathbb{Z}/(q_{1l}^{g_{1l}}) \oplus \mathbb{Z}/(p_1^{e_1 - 1}) \oplus \cdots$$
$$\cdots \oplus \mathbb{Z}/(2^{f_k}) \oplus \mathbb{Z}/(q_{k1}^{g_{k1}}) \oplus \cdots \oplus \mathbb{Z}/(q_{kl}^{g_{kl}}) \oplus \mathbb{Z}/(p_k^{e_k - 1})$$

no qual notamos que

$$(\frac{p_1 - 1}{2}, 0, \ldots, \frac{p_k - 1}{2}, 0) \mapsto (2^{f_1 - 1}, 0, \ldots, 0, \ldots, 2^{f_k - 1}, 0, \ldots, 0)$$

Sejam G_0 e G_1 como na demonstração do teorema 13. Lembramos que $(\mathbb{Z}(N))^\times = G_0 \oplus G_1$ onde G_0 tem ordem $2^{(f_1 + \cdots + f_k)}$ e G_1 tem ordem ímpar. Sem perda de generalidade, façamos $0 < f_1 \leq f_2 \leq \ldots \leq f_k$.

Notamos que

$$-1 \mapsto (2^{f_1 - 1}, \ldots, 2^{f_k - 1}; 0, \ldots, 0) = (s; 0)$$

onde $s = (2^{f_1 - 1}, \ldots, 2^{f_k - 1}) \in G_0$ tem ordem 2.

Ora, o sorteio aleatório de m em $(\mathbb{Z}/(N))^\times$ com distribuição uniforme de probabilidades, conforme o enunciado deste teorema, se encontra em cor-

respondência bijetora com o sorteio aleatório de (a, b) em $G_0 \oplus G_1$. Já neste segundo sorteio somente nos interessa o sorteio aleatório de a em G_0, dado que G_1 tem ordem ímpar.

Existem $2^k - 1$ elementos de ordem 2 em G_0. Esses elementos são da forma $(a_1, \ldots, a_k)$, com $a_i \in \{0, 2^{f_i-1}\}, i = 1, \ldots, k, (a_1, \ldots, a_k) \neq (0, \ldots, 0)$.

Seja $a = (a_1, a_2, \ldots, a_k)$ um elemento aleatório de G_0.

Seja $2^{j_i} = \text{ord}_{\mathbb{Z}/(2^{f_i})}\, a_i, i \in \{1, \ldots, k\}$. Assim $j_i \in \{0, \ldots f_i\}$.

Observamos que $\text{ord}_{G_0}\, a = 2^j$ onde $j = \text{máximo}\{j_1, \ldots, j_k\}$ pois $\text{ord}_{G_0}\, a = \text{mmc}(\text{ord}_{\mathbb{Z}/(2^{f_1})}\, a_1, \ldots, \text{ord}_{\mathbb{Z}/(2^{f_k})}\, a_k)$. Notamos também que $j = 0 \Longleftrightarrow a = (0, \ldots, 0)$.

Para cada valor positivo de j vamos verificar quantos elementos a de G_0 satisfazem $\text{ord}_{G_0}\, a = 2^j$ e $2^{j-1}a = s$.

Temos $\text{ord}_{\mathbb{Z}/(2^{f_i})}\, a_i = 2^{j_i}, j = \text{máx}(j_1, \ldots, j_k)$ e afirmamos:

Afirmação: $2^{j-1}a = s \Longleftrightarrow j_1 = j_2 = \ldots = j_k = j \in \{1, \cdots, f_1\}$.

Com efeito, temos $2^{j-1}a = s \Longleftrightarrow 2^{j-1}a_i = 2^{f_i-1}, i \in \{1, \ldots, k\} \Longleftrightarrow \text{ord}_{\mathbb{Z}/(2^{f_i})}\, a_i = 2^j, i \in \{1, \ldots, k\} \Longleftrightarrow 2^j = 2^{j_i}, \in \{1, \ldots, k\} \Longleftrightarrow j = j_i, i \in \{1, \ldots, k\}$. E mais, com elementos a de G_0 satisfazendo $\text{ord}_{G_0}\, a = 2^j$ e $2^{j-1}a = s$ tem-se $j \in \{1, \ldots, f_1\}$, pois $j = \text{máx}(j_1, \ldots, j_k), 0 < f_1 \leq f_2 \leq \ldots \leq f_k$ e $j_i \in \{0, \ldots f_i\}, i \in \{1, \ldots, k\}$.

Para verificar quantos elementos de G_0 satisfazem a condição $j_1 = j_2 = \ldots = j_k = j \in \{1, \cdots, f_1\}$, lembramos a lista dos elementos a de $\mathbb{Z}/(2^f)$ segundo sua ordem, apresentada na seção 2.3, e observamos que:

se $j = 1$ temos um elemento (o próprio s).

se $j = 2$ temos 2^k elementos, que são da forma $(a_1, \ldots, a_k)$, com $a_i \in \{2^{f_i-2}, 3 \cdot 2^{f_i-2}\}, i = 1, \ldots, k$.

se $j = 3$ temos $(2^k)^2$ elementos, que são da forma $(a_1, \ldots, a_k)$, com $a_i \in \{2^{f_i-3}, 3 \cdot 2^{f_i-3}, 5 \cdot 2^{f_i-3}, 7 \cdot 2^{f_i-3}\}, i = 1, \ldots, k$.

$\ldots$

se $j = f_1$ temos $(2^k)^{f_1-1}$ elementos.

Portanto existem $\sum_{i=0}^{f_1-1}(2^k)^i = \frac{2^{kf_1}-1}{2^k-1}$ elementos de G_0 que satisfazem a condição $j_1 = j_2 = \ldots = j_k = j \in \{1, \cdots, f_1\}$.

Tomamos um elemento aleatório $a = (a_1, \ldots, a_k)$ uniformemente distribuído em G_0.

Seja $\text{ord}_{G_0}\, a = 2^j$.

Temos

$$\text{Prob}((j > 0) \wedge (2^{j-1} \cdot a = s)) = \frac{2^{kf_1} - 1}{(2^k - 1) \cdot 2^{f_1 + \cdots + f_k}}$$

Notando que $f_1 + \cdots + f_k \geq kf_1$, temos

$$\text{Prob}((j > 0) \wedge (2^{j-1} \cdot a = s)) < \frac{1}{2^k - 1}$$

E o teorema está demonstrado. □

Corolário 7. *Seja $N \in \mathbb{N}$ ímpar com pelo menos dois fatores primos distintos. Sorteia-se aleatoriamente $m \in (\mathbb{Z}/(N))^\times$, com distribuição uniforme de probabilidades. Então*

$$\text{Prob}((\text{ord}_N\, m \text{ ser par}) \wedge (m^{\frac{\text{ord}_N\, m}{2}} + 1 \equiv 0 \pmod N)) < \frac{1}{3}$$

Os teoremas 13 e 14 dizem respeito a limitantes superiores para as probabilidades de ocorrência de dois tipos de fracasso no algoritmo de fatoração de Shor.

Vamos encerrar esta seção com um teorema que sintetiza seus resultados, indicando um limite inferior para a probabilidade de o algoritmo *não incorrer* nos dois tipos de fracasso mencionados.

Teorema 15. *Seja $N \in \mathbb{N}$ ímpar com k fatores primos distintos. Sorteia-se aleatoriamente $m \in (\mathbb{Z}/(N))^\times$, com distribuição uniforme de probabilidades. Então*

$$\text{Prob}((\text{ord}_N\, m \text{ ser par}) \wedge (m^{\frac{\text{ord}_N\, m}{2}} + 1 \not\equiv 0 \pmod N)) > 1 - \frac{2^{k+1} - 1}{2^k(2^k - 1)}$$

Demonstração. Consideremos os eventos:

$$A : \text{ord}_N\, m \text{ é par}$$

$$B : m^{\frac{\text{ord}_N\, m}{2}} + 1 \equiv 0 \pmod N$$

$$C : m^{\frac{\text{ord}_N\, m}{2}} + 1 \not\equiv 0 \pmod N$$

Do teorema 13 temos imediatamente $\text{Prob}(A) \geq 1 - \frac{1}{2^k}$.

Já do teorema 14 sabemos que

$$\text{Prob}(A \wedge B) < \frac{1}{2^k - 1}.$$

Temos então

$$\text{Prob}(A) = \text{Prob}(A \wedge B) + \text{Prob}(A \wedge C) \geq 1 - \frac{1}{2^k}.$$

E podemos escrever

$$\frac{1}{2^k - 1} + \text{Prob}(A \wedge C) > 1 - \frac{1}{2^k}.$$

Logo

$$\text{Prob}(A \wedge C) > 1 - \frac{2^{k+1} - 1}{2^k(2^k - 1)}$$

E o teorema está demonstrado □

Corolário 8. *Seja $N \in \mathbb{N}$ ímpar com pelo menos dois fatores primos distintos. Sorteia-se aleatoriamente $m \in (\mathbb{Z}/(N))^\times$, com distribuição uniforme de probabilidades. Então*

$$\text{Prob}((\text{ord}_N\, m \text{ ser par}) \wedge (m^{\frac{\text{ord}_N\, m}{2}} + 1 \not\equiv 0 \pmod N)) > \frac{5}{12}$$

Com este último resultado temos um limitante inferior próximo a $\frac{1}{2}$ para a probabilidade de sucesso do algoritmo de Shor em sua parte clássica. A probabilidade de sucesso na parte quântica do algoritmo está associada a um processo de medição física e será analisada nos capítulos 5 e 6.

2.11 Notação Assintótica

Um aspecto central nesta dissertação é a avaliação da eficiência do algoritmo de fatoração de Shor, comparada à eficiência da algoritmos clássicos de fatoração.

Um dos procedimentos mais adotados para isso é a notação assintótica, que é útil para avaliar o comportamento de uma função nos inteiros positivos

$$f : \mathbb{N}^* \to \mathbb{N}^* \tag{2.1}$$

$$n \mapsto f(n) \tag{2.2}$$

para valores grandes de n e se aplica diretamente para informar em essência quantos passos leva um algoritmo para ser rodado.

Assim, voltando para o objeto de estudo desta dissertação, veremos que a eficiência do algoritmo de Shor pode ser avaliada aplicando-se a notação assintótica. Ela irá nos indicar como o número de operações requeridas para execução do algoritmo cresce com o número n de dígitos da representação em base 2 do número N a ser fatorado, com $n = \lceil \log_2 N \rceil$ onde $\lceil x \rceil$ denota o único inteiro k tal que $k - 1 < x \leq k$. O mesmo pode ser feito para algoritmos clássicos, propiciando assim a comparação desejada.

A notação assintótica dispõe de um recurso que fornece um embasamento preciso a essa avaliação: a notação "O", que é usada para estabelecer limites superiores para o comportamento assintótico de uma função.

Sejam $f(n)$ e $g(n)$ duas funções nos inteiros positivos. Dizemos que $f(n)$ pertence à classe de funções $O(g(n))$, ou mais simplesmente que $f(n)$ é $O(g(n))$, se existirem c real e n_0 inteiro tais que para todo $n > n_0$ temos $f(n) \leq cg(n)$. A notação "O" é útil para estudar o comportamento de pior caso de um algoritmo.

Após esta incursão em aspectos de teoria dos números relevantes para o tema desta dissertação, no próximo capítulo iniciaremos a apresentação dos conceitos básicos de mecânica quântica aplicados à computação quântica.

3
Breve Introdução à Mecânica Quântica Voltada para a Computação Quântica

3.1
Do bit ao qubit

Ao declarar que em um computador nenhum fenômeno ou processo pode violar as leis da Mecânica Quântica Feynman [2] apontou para o óbvio que até então não era visto e com seu artigo seminal [7] abriu caminho para a síntese de duas das maiores inovações científicas do século XX: a Mecânica Quântica e as Ciências da Computação.[1]

A partir desse insight de Feynman teve início um amplo esforço de pesquisa que pode eventualmente levar à superação de conceitos da computação clássica fundamentados na máquina universal de Turing. Entende-se aqui "superar" em sentido dialético hegeliano (aufheben), que inclui conservar (aufbewahren), elevar (erhören) e vencer (überwinden). Em mesmo sentido entendemos que a Mecânica Quântica superou a Física Newtoniana. Assim, as leis do movimento de Newton continuam sendo importantes em Física, assim como a máquina de Turing continua a ser um conceito muito importante na teoria de computação.

A aplicação de conceitos da Mecânica Quântica às Ciências da Computação, em consonância com a sugestão original de Feynman, se inicia em nível da "partícula elementar" das ciências da computação, o bit, que pode ser substituído ou ressignificado pelo qubit.

Com efeito, enquanto no contexto da computação clássica um bit é representado pelos inteiros 0 e 1, a introdução do qubit em Computação Quântica se dá a partir de uma abordagem estritamente físico-quântica do computador, como veremos a seguir.

[1] A elaboração deste capítulo baseou-se principalmente em Lavor et alli [34] e em Nielsen e Chuang [35].

3.2 Os Postulados da Mecânica Quântica

A Mecânica Quântica é uma estrutura matemática para o desenvolvimento de teorias aplicáveis a sistemas, processos e fenômenos físicos — como por exemplo o próprio computador e a computação. Por si mesma a Mecânica Quântica não prescreve quais leis um sistema físico específico deve obedecer, mas provê uma estrutura conceitual e matemática para o desenvolvimento de tais leis.

A Mecânica Quântica baseia-se em quatro postulados, que estabelecem a conexão entre o mundo físico e o formalismo matemático, a saber:

Postulado 1: A todo sistema físico isolado está associado um espaço vetorial complexo com produto interno (um espaço de Hilbert) conhecido como o espaço de estado do sistema. O sistema é completamente descrito por seu vetor de estado, que é um vetor unitário em seu espaço de estado.

Postulado 2: A evolução de um sistema físico isolado é descrita pela equação de Schrödinger:

$$i\frac{h}{2\pi}\frac{d|\psi\rangle}{dt} = H|\psi\rangle$$

onde h é a constante de Planck e H é o operador hamiltoniano do sistema.

A resolução desta equação diferencial permite relacionar o estado do sistema em um instante t_1 com seu estado no instante t_2 mediante

$$|\psi_2\rangle = U|\psi_1\rangle$$

onde U é um operador linear unitário que depende somente de t_1 e t_2, $|\psi_1\rangle$ é o vetor de estado no instante t_1 e $|\psi_2\rangle$ é o vetor de estado no instante t_2.

Postulado 3: Medidas quânticas são descritas por um conjunto $\{M_m\}$ de operadores de medida atuantes no espaço de estado associado ao sistema submetido à medida. O resultado da medida é denotado pelo subíndice m. Se o sistema se encontrar no estado $|\psi\rangle$ imediatamente antes da medida, então a probabilidade de que o resultado m ocorra é dada por

$$\text{Prob}(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$$

onde $M_m^\dagger$ é o operador adjunto de M_m, M_m é projetor e o estado do sistema

após a medida será dado por

$$|\phi\rangle = \frac{M_m|\psi\rangle}{\sqrt{\text{Prob}(m)}}$$

Postulado 4: O espaço de estado de um sistema físico composto é o produto tensorial dos espaços de estado dos sistemas físicos componentes.

Ao longo da próxima seção a conexão entre o mundo físico e o formalismo matemático, propiciada pela Mecânica Quântica, será apresentada tendo o computador como a entidade física de referência.

3.3 O Computador visto como um Sistema Físico-Quântico

Como qualquer outro sistema físico, também o computador deve atender aos postulados da Mecânica Quântica e é interessante visualizar passo a passo como isso se verifica, lançando mão do formalismo propiciado pela Álgebra Linear.

Na perspectiva do postulado 1, o qubit nada mais é que o sistema mecânico quântico mais simples, associado a um espaço de Hilbert bidimensional — e pode se encontrar fisicamente tanto em um computador como em qualquer outro ambiente natural ou construído pelo homem.

Já de acordo com o postulado 2, o qubit pode evoluir mediante transformações lineares unitárias. Em particular no computador tal evolução corresponderá ao processamento computacional quântico, qubit a qubit, que vai caracterizar a descrição de algoritmos quânticos.

Voltando ao postulado 1, o qubit é representado por um vetor de estado ou, em linguagem simplificada, por um "estado", que pode ser denotado como $|0\rangle$, $|1\rangle$[2] ou como uma combinação linear desses dois estados com coeficientes complexos:

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

Nesse contexto $|\psi\rangle$ é dito uma "superposição" dos estados $|0\rangle$ e $|1\rangle$.

Os dois estados representados por $|0\rangle$ e $|1\rangle$ constituem uma base ortonormal, denominada base computacional do espaço de Hilbert bidimensional,

[2]Trata-se aqui da notação de Dirac ou "notação dos físicos", de uso universal em Mecânica Quântica.

e são representados em forma matricial por [3]

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Cabe aqui notar que não há bijeção entre o qubit e o vetor que o representa. Com efeito, multiplicando-se α e β em $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ por um número complexo unimodular, o estado não muda.[4]

Uma diferença marcante entre a Mecânica Quântica e as ciências clássicas da computação já se apresenta aqui e terá consequências consideráveis em aplicações práticas: enquanto o bit somente assume dois valores possíveis, o qubit pode armazenar quantidade muito maior de informação nos coeficientes α e β. Contudo essa informação reside em nível quântico e para levá-la ao nível clássico é necessário medir o qubit.

Uma opção frequentemente usada é a medida do qubit na base computacional $(|0\rangle, |1\rangle)$ do espaço de Hilbert. Segue-se do postulado 3 que o processo de medida perturba inevitavelmente o qubit provocando a transformação do estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ em um dos estados da base correspondente. Em outras palavras, com a medida do qubit na base computacional, o processo leva o vetor de estado $|\psi\rangle$ a "colapsar" em um dos estados $|0\rangle$ ou $|1\rangle$ com probabilidades $|\alpha|^2$ e $|\beta|^2$ respectivamente.

É impossível conhecer os valores de α e β antes da medida. Podemos estimar $|\alpha|^2$ e $|\beta|^2$ mediante muitas medidas do mesmo estado inicial $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Esse estado deve ser preparado (reconstituído) antes de cada uma dessas medidas. Ao realizarmos a primeira medida a partir de um certo estado inicial, temos como resultado (output) um dos vetores de estado da base computacional, resultante do colapso do estado inicial previsto no postulado 3. Se depois disso realizarmos uma segunda medida, agora sobre este output, obteremos como resultado novamente este mesmo estado, pois o output se manterá inalterado após o primeiro colapso e assim sucessivamente. Por essa razão é necessário preparar (reconstituir) o estado original, antes de cada uma das muitas medidas.

Para avançar além da mera estimativa de $|\alpha|^2$ e $|\beta|^2$ e conhecer a fase relativa de α e β (supondo ambos diferentes de 0) é necessário medir em bases outras que a base computacional.

[3] Os dois elementos constituintes da base computacional (qualquer que seja ela) são frequentemente denotados como $|0\rangle$ e $|1\rangle$. Já a opção de representá-los matricialmente é arbitrária. Contudo essa opção será mantida em todo o texto desta dissertação pois é a mais usual.

[4] Por exemplo, $\frac{(i|0\rangle - |1\rangle)}{\sqrt{2}}$ e $\frac{|0\rangle + i|1\rangle)}{\sqrt{2}}$ representam o mesmo qubit.

Da Álgebra Linear sabemos que uma transformação linear U em um espaço de Hilbert é *unitária* se e somente se $U^\dagger U = UU^\dagger = I$, onde I é o operador identidade. Da Mecânica Quântica sabemos que, mesmo se nenhuma medida for tomada, estados podem ser modificados por operadores unitários. Nestas condições, um qubit $|\psi\rangle = \alpha_0|0\rangle + \beta_0|1\rangle$ pode ser modificado para $U|\psi\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$ com $|\alpha_1|^2 + |\beta_1|^2 = 1$.

Os vetores de estado $|0\rangle$, $|1\rangle$ ou em geral $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ são suficientes para se lidar com um qubit. Entretanto, para a execução de qualquer cálculo útil é necessário lidar com mais de um qubit — o que requer a introdução do conceito de produto tensorial, que veremos em seguida.

Sejam V e W espaços vetoriais complexos de dimensões m e n, respectivamente. Define-se produto tensioral $V \otimes W$ como o espaço vetorial complexo de dimensão mn tal que os elementos de $V \otimes W$ são combinações lineares de produtos da forma $|v\rangle \otimes |w\rangle$, com $|v\rangle \in V$, $|w\rangle \in W$ [5].

O produto tensorial satisfaz as seguintes propriedades, para $z \in \mathbb{C}, |v\rangle, |v_1\rangle, |v_2\rangle \in V, |w\rangle, |w_1\rangle, |w_2\rangle \in W$:

- $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$
- $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = (|v_1\rangle \otimes |w\rangle) + (|v_2\rangle \otimes |w\rangle)$
- $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = (|v\rangle \otimes |w_1\rangle) + (|v\rangle \otimes |w_2\rangle)$

Dados dois operadores lineares A, B e os espaços vetoriais complexos V, V', W, W', de dimensões m, m', n, n' respectivamente, tais que

$$A : V \to V', B : W \to W'$$

define-se o operador $A \otimes B$ tal que

$$(A \otimes B) : (V \otimes W) \to (V' \otimes W')$$

$$|v\rangle \otimes |w\rangle \to A|v\rangle \otimes B|w\rangle$$

[5] Também se usam as notações simplificadas $|v\rangle|w\rangle$, $|v, w\rangle$ ou $|vw\rangle$. Cabe notar que a definição de produto tensorial como espaço vetorial permite nele abrigar todos os vetores de estado resultantes de operações com qubits, típicas da Computação Quântica, satisfazendo assim a prescrição expressa do postulado 1.

Uma representação matricial de $A \otimes B$ é dada por

$$\begin{bmatrix} A_{11}B & ... & A_{1m'}B \\ . & . & . \\ . & . & . \\ . & . & . \\ A_{m1}B & ... & A_{mm'}B \end{bmatrix}$$

onde A é $m \times m'$ e B é $n \times n'$ e adotamos a ordem lexicográfica para ordenação das linhas e das colunas.

Outra representação matricial de $A \otimes B$ é dada por

$$\begin{bmatrix} AB_{11} & ... & AB_{1n'} \\ . & . & . \\ . & . & . \\ . & . & . \\ AB_{n1} & ... & AB_{nn'} \end{bmatrix}$$

onde A é $m \times m'$, B é $n \times n'$ e adotamos a ordem antilexicográfica para ordenação das linhas e das colunas.

Neste texto vamos adotar a ordem lexicográfica.

A representação matricial pode ser aplicada ao produto tensorial dos vetores da base computacional do espaço de Hilbert bidimensional, em nossa opção $|0\rangle$ e $|1\rangle$, relacionada a um qubit.

Em um computador com dois qubits, onde cada qubit pode estar no estado $|0\rangle$, no estado $|1\rangle$ ou em uma combinação linear complexa desses dois estados, há quatro produtos tensoriais a considerar, que irão constituir a base computacional desse computador: $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Esses quatro estados, também representados como $|0\rangle, |1\rangle, |2\rangle, |3\rangle$ respectivamente, constituem a base computacional para o espaço de Hilbert tetradimensional onde residem os estados do computador de dois qubits. [6]

Assim, enquanto um qubit é representado pelo vetor de estado $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$, dois qubits são representados pelo vetor de estado $|\varphi\rangle$, uma superposição dos quatro estados da base computacional do espaço de Hilbert tetradimensional, denotada como

Notação 1. $|\varphi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle = \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle + \delta|3\rangle$.

[6] A relação entre ambas as representações é óbvia: enquanto na primeira representação os labels dos estados da base computacional são representações dos quatro primeiros números naturais em base 2, na segunda representação estes se encontram representados em base 10.

Finalmente, podemos generalizar para a representação de n qubits mediante o vetor de estado $|\psi\rangle$, uma superposição dos 2^n estados da base computacional do espaço de Hilbert 2^n-dimensional, $|0\rangle, |1\rangle, ..., |2^n - 1\rangle$:

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

sujeito à restrição $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$, já que, segundo o postulado 3, uma medida na base computacional do estado $|\psi\rangle$ leva este estado a colapsar no estado $|j\rangle$ com probabilidade $|\alpha_j|^2$, $0 \leq j \leq 2^n - 1$. Essa medida é feita qubit a qubit, fornecendo valores 0 ou 1 para cada qubit medido, os quais formam a expansão de j na base 2.[7]

Como o estado do computador quântico é representado por um vetor em um espaço de Hilbert 2^n-dimensional, se o número de qubits aumentar linearmente, a dimensão do espaço vetorial aumentará exponencialmente. Assim, basta acrescentar um quibt em um computador quântico para que se dobre a dimensão do espaço de Hilbert a ele associado.

3.4 O Emaranhamento, Entidade Central da Computação Quântica

O produto tensorial por si só ainda é insuficiente para representar todo o espaço de Hilbert 2^n-dimensional.

Por exemplo, facilmente se verifica que nem sempre um vetor de estado em um espaço de Hilbert tetradimensional pode ser representado como produto tensorial de dois qubits. Essa insuficiência é uma decorrência imediata do fato de que o mapa

$$\otimes : \mathbb{C}^2 \times \mathbb{C}^2 \longrightarrow \mathbb{C}^2 \otimes \mathbb{C}^2$$

$$(v, w) \longrightarrow v \otimes w$$

não é sobrejetor.

Com efeito, sejam $v = a|0\rangle + b|1\rangle$ e $w = c|0\rangle + d|1\rangle$ dois vetores de estado em espaços de Hilbert bidimensionais. Temos $v \otimes w = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$. Comparando esta expressão com a forma geral de um vetor de estado em um espaço de Hilbert tetradimensional dada pela notação 1 concluímos que tal vetor de estado somente poderá ser representado como produto tensorial

[7] Notamos que a ordem segundo a qual as medidas são tomadas qubit a qubit é irrelevante, já que a base computacional é constituída por vetores de estado de um produto tensorial e os sucessivos resultados das medidas correspondem à aplicação sucessiva dos operadores $P_0, P_1, ..., P_{n-1}$ que comutam entre si.

de dois qubits se

$$\det \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = 0$$

Sob a perspectiva da Mecânica Quântica, o mesmo fato pode ser verificado considerando-se um computador quântico de dois qubits como um sistema físico. Se nele tivermos um qubit no estado $|\phi\rangle = a|0\rangle + b|1\rangle$ e outro qubit no estado $|\psi\rangle = c|0\rangle + d|1\rangle$ então, segundo o postulado 4, o estado do computador quântico será o produto tensorial

$$|\phi\rangle \otimes |\psi\rangle = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle$$

Novamente comparando esta expressão com a forma geral do estado de um computador de dois qubits dada pela notação 1 concluímos que o estado geral de um computador de dois qubits, ao qual se deve associar um espaço de Hilbert tetradimensional (segundo o postulado 1), não é necessariamente o produto tensorial de dois estados de um qubit.

A extensão para espaços de Hilbert n-dimensionais com $n \geq 3$ é análoga e decorre do fato de que o mapa

$$\otimes : \mathbb{C}^n \times \mathbb{C}^n \longrightarrow \mathbb{C}^n \otimes \mathbb{C}^n$$

$$(v, w) \longrightarrow v \otimes w$$

não é sobrejetor.

Com efeito, seja $(e_1, \ldots, e_n)$ uma base de $\mathbb{C}^n$.

Assim $\mathbb{C}^n \otimes \mathbb{C}^n$ tem base $(e_1 \otimes e_1, e_1 \otimes e_2, \ldots, e_1 \otimes e_n, e_2 \otimes e_1, e_2 \otimes e_2, \ldots, e_2 \otimes e_n, \ldots, e_n \otimes e_1, e_n \otimes e_2, \ldots, e_n \otimes e_n)$.

No espaço de Hilbert $\mathbb{C}^n \times \mathbb{C}^n = \mathbb{C}^{n^2}$ um elemento típico $|\tau\rangle$ é representado em geral por

Notação 2. $|\tau\rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{ij} e_i \otimes e_j$.

Tomemos agora dois elementos típicos $v = \sum_{i=1}^{n} u_i e_i$, $w = \sum_{j=1}^{n} w_j e_j$ em $\mathbb{C}^n$.

Temos $v \otimes w = \sum_{i=1}^{n} \sum_{j=1}^{n} u_i v_j e_i \otimes e_j$.

Comparando este resultado com a forma geral de representação de um elemento típico de $\mathbb{C}^{n^2}$ dada pela notação 2 verificamos que tal elemento somente poderá ser representado por um produto tensorial se

$$\det \begin{bmatrix} \alpha_{ii} & \alpha_{ij} \\ \alpha_{ji} & \alpha_{jj} \end{bmatrix} = 0$$

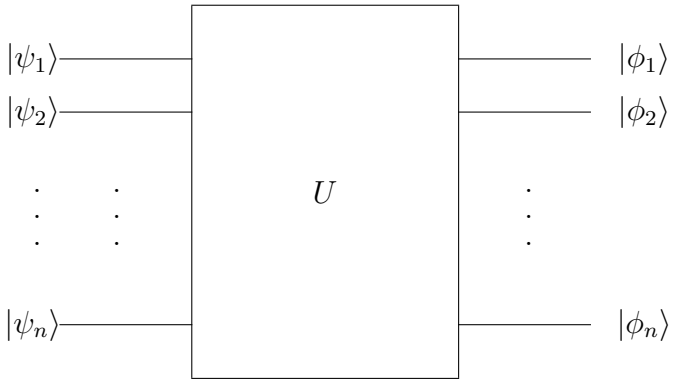

Figura 3.1: Esboço do desenho de um computador quântico

para todo $i, j \in \{1, \ldots, n\}$. Portanto o estado geral de um computador de n qubits não é necessariamente um produto tensorial.

Estados de dois ou mais qubits que não são produtos tensoriais são denominados estados *emaranhados*. Estados emaranhados têm um papel destacado em computação quântica. Para descrever um estado emaranhado precisamos de um número de parâmetros que cresce exponencialmente com o número de qubits. O simples acréscimo de um qubit em um computador quântico duplica a dimensão do espaço de Hilbert a ele associado.

3.5 Um Singelo Esboço de um Computador Quântico

Após esta breve revisão, podemos esboçar o desenho de um computador quântico. Na figura 3.1 consideramos um input não emaranhado constituído por n estados $|\psi_1\rangle$, $|\psi_2\rangle$, ..., $|\psi_n\rangle$, onde cada $|\psi_i\rangle$ é $|0\rangle$ ou $|1\rangle$. Nela o output é em geral um estado emaranhado.

O esboço apresentado na figura 3.1 será retomado e representado de forma mais detalhada no capítulo 7, mediante o uso de circuitos constituídos por *portas* articuladas. Por ora tal esboço deve permanecer como caixa preta. Contudo desde já admitimos que em um computador assim tão singelamente representado se processa a evolução de sistemas físicos, os qubits, segundo prescrito no postulado 2.

Portas quânticas são representações de operadores lineares unitários reversíveis em circuitos: este é um fato decisivo para a interface entre Mecânica Quântica e Computação Quântica. Somente operadores unitários podem ser implementados e executados no laboratório de forma determinista. Operadores não unitários somente podem ser implementados e executados probabilisticamente e tal implementação e execução somente pode ser feita com portas unitárias juntamente com medidas.

No capítulo 7 veremos como qualquer operador pode ser representado por um conjunto articulado (um "circuito") composto por portas de um qubit de apenas três tipos distintos e por portas de dois qubits de um único tipo e definiremos o conceito de portas universais.

Assim como no modelo clássico da máquina de Turing, também aqui o último passo é constituído pela medida do estado final — a se realizar de acordo com o prescrito no postulado 3. Em certos casos isso pode ser feito medindo cada qubit individualmente, o que retorna zeros ou uns que formam o resultado final do cálculo quântico. Em outros casos medem-se vários qubits em conjunto em espaços de Hilbert de dimensão maior que dois.

Concluída esta breve apresentação de conceitos básicos de Mecânica Quântica aplicáveis à Computação Quântica, cabe agora dar início a uma tentativa de "costura" dessas duas ciências, por meio de duas representações do algoritmo de Shor: como circuito e como algoritmo passo a passo.

4
Circuitos Quânticos: Principais Portas e sua Implementação

4.1
As Principais Portas

Um computador clássico universal pode ser representado de diversas formas, das quais a mais consagrada é a máquina universal de Turing. Contudo, nesta dissertação adotaremos uma representação equivalente, qual seja, a dada por um circuito composto por elementos primitivos interconectados [1].

Pode-se tomar esses elementos primitivos como sendo apenas dois, as portas NOT e AND, com um input e com dois inputs respectivamente[2]. A transposição da porta NOT para o caso quântico é apresentada na figura 4.1, onde X é o operador unitário representado em forma matricial por[3]

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

O operador X permuta os vetores da base computacional: se o input $|\psi\rangle$ for $|0\rangle$, o output será $|1\rangle$ e vice-versa. Já se o input for uma superposição dos vetores da base computacional $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, o output será $\alpha|1\rangle + \beta|0\rangle$, um caso que não tem contrapartida em computação clássica.

A porta X não é a única porta de um qubit em circuitos quânticos. Há infinitas portas de um qubit, pois há infinitos operadores unitários.

Em particular, a matriz X é uma das três matrizes de Pauli, fundamentais em computação quântica, que formam uma base para o espaço vetorial das

[1] Para uma prova dessa equivalência, cf. Nielsen e Chuang [35], pp. 134-135. Na elaboração deste capítulo nos baseamos principalmente em Lavor et alli [34].

[2] Cf. Feynman [7].

[3] A figura 4.1 apresenta nosso primeiro exemplo de uma porta de um qubit que poderá compor circuitos quânticos. Doravante utilizaremos indistintamente os termos "operador unitário" e "porta", o primeiro proveniente da álgebra linear e o segundo da computação clássica e da eletrônica digital.

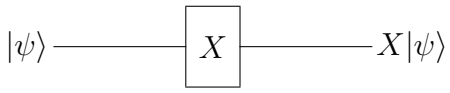

Figura 4.1: Porta X

matrizes hermitianas 2×2 de traço nulo:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

As matrizes de Pauli, juntamente com a matriz identidade 2×2, formam uma base para o espaço vetorial das matrizes hermitianas 2×2.

Entre as portas de um qubit destacam-se ainda por suas inúmeras aplicações em computação quântica a porta de Hadamard e o operador rotacional.

A porta de Hadamard é representada em forma matricial por

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{\sqrt{2}}(X + Z)$$

A aplicação da porta de Hadamard sobre os vetores da base computacional resulta em

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

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

Notamos que os outputs da porta de Hadamard para os inputs $|0\rangle$ e $|1\rangle$ são superposições dos estados da base computacional com igual amplitude (igual peso).

Para uma porta de Hadamard de dois qubits aplicada ao estado $|0\rangle^{\otimes 2} = |0\rangle \otimes |0\rangle$ com 2 qubits temos:

$$H^{\otimes 2}|0\rangle|0\rangle = (H \otimes H)(|0\rangle \otimes |0\rangle) = H|0\rangle \otimes H|0\rangle =$$

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

Com os labels dos estados representados em base decimal temos $H^{\otimes 2}|0\rangle|0\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$. Novamente o output é uma superposição dos estados da base computacional com igual amplitude (igual peso).

No caso geral de uma porta de n qubits temos:

$$H^{\otimes n}|0\rangle^{\otimes n} = H^{\otimes n}|0, ..., 0\rangle = (H|0\rangle)^{\otimes n} = (\frac{|0\rangle + |1\rangle}{\sqrt{2}})^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

onde novamente representamos os labels dos estados em base decimal.

Assim temos, no caso geral, que o produto tensorial de n operadores de Hadamard produz uma superposição de todos os estados da base computacional com igual amplitude (igual peso), quando aplicado ao estado $|0\rangle^{\otimes n}$ com n qubits.

Já o operador rotacional é representado pela matriz

$$\mathcal{R}_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \text{ com } \phi \in \mathbb{R}.$$

Dizemos que este operador exerce rotação em torno do eixo z.

Entre os infinitos operadores rotacionais em torno do eixo z cabe destacar os operadores S ("fase") e T ("$\frac{\pi}{8}$") [4].

$$S = \mathcal{R}_{\frac{\pi}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad T = \mathcal{R}_{\frac{\pi}{4}} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$$

Entre as portas quânticas de dois qubits destaca-se a porta CNOT, que tem dois qubits no input e dois qubits no output. No input, o primeiro qubit atua como controlador e o segundo como alvo. O operador CNOT atua de forma que o segundo qubit será trocado se e somente se o primeiro qubit for $|1\rangle$. O primeiro qubit se mantém inalterado. Portanto temos:

$$|00\rangle \mapsto |00\rangle$$

$$|01\rangle \mapsto |01\rangle$$

$$|10\rangle \mapsto |11\rangle$$

$$|11\rangle \mapsto |10\rangle$$

[4] A denominação "$\frac{\pi}{8}$" para o operador T se deve a razões históricas (cf. Nielsen e Chuang [35], p. 174).

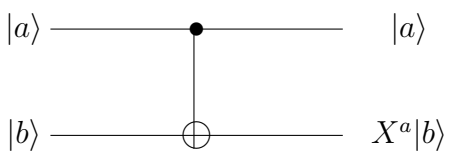

Figura 4.2: Porta CNOT

E a representação matricial do operador da porta CNOT será

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

A figura 4.2 apresenta a porta CNOT, onde $|a\rangle$ é o primeiro qubit do input (qubit "controlador", $|b\rangle$ é o segundo qubit do input (qubit "alvo"), $X^1 = X$ é o operador NOT e $X^0 = I$ é o operador identidade. Ela é válida para os casos particulares $|a\rangle = |0\rangle$ ou $|a\rangle = |1\rangle$.

No caso geral, tal representação não faria sentido e poderia induzir a erro de interpretação. Com efeito, o intput pode não ser vetor produto mas um emaranhamento de elementos da base computacional $\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$ ou mesmo, no caso mais geral, um estado misto. Se a porta CNOT for parte de um circuito maior, o input poderia ser uma superposição de estados dessa forma tensoriados com outros no restante do produto tensorial de espaços de Hilbert ou ainda, no caso mais geral, um estado misto.

A porta CNOT e as portas de Hadamard, fase e $\frac{\pi}{8}$ formam um *conjunto universal de portas*, isto é, qualquer outra porta, operando sobre dois ou mais qubits, pode ser aproximada arbitrariamente bem por uma porta constituída por um conjunto articulado dessas quatro portas. [5] Isto tem um significado de extrema importância em computação quântica, pois permite representar o computador quântico como um circuito contendo somente essas portas. Além disso, a decomposição de qualquer porta em portas elementares corresponde à ordem natural dos comandos passo a passo que descrevem algoritmos.

Desta maneira a computação quântica com a representação por meio de

[5] Cf. Nielsen e Chuang [35], pp. 191-197.

circuito permite visualizar algoritmos quânticos "fisicamente", como sequência de operadores (portas) em paralelo ou em série (circuitos) que atuam sobre conjuntos de qubits.

4.2
Implementação

4.2.1
Um Breve Review

Ainda que seja impossível estimar o quanto estamos distantes da realização física de um computador quântico com desempenho significativo, nas últimas décadas tem havido notável empenho tendo em vista atingir essa meta.

Em nível de pesquisa acadêmica, Feynman [7] sugere passos para a implementação de computadores quânticos, com considerações em nível de lógica, usando as portas NOT e CNOT [6]. Contudo Feynman se atém à representação e ao tratamento logico-matemático dos circuitos, sem abordar o modo de sua construção física.

Em nível de implementação física efetiva de computadores quânticos com íons aprisionados, avanços experimentais recentes podem ser encontrados em Haffner et alli [36]. Em particular várias implementações de qubits, portas quânticas e algumas experiências-chave são discutidas. Além disso, os autores analisam algumas implementações de algoritmos quânticos, como teletransporte determinístico de informações quânticas, bem como um esquema de correção de erros.

Já em M. H. Devoret, J. M. Martinis e A. Allraff [37] encontram-se recentes contribuições à implementação física de computadores quânticos com circuitos supercondutores (atualmente considerada a técnica mais promissora). Este método baseia-se em qubits supercondutores, que são circuitos elétricos de estado sólido fabricados usando técnicas emprestadas de circuitos integrados convencionais. Eles são baseados na junção de túnel Josephson, o único elemento de circuito não dissipativo fortemente não linear disponível a baixa temperatura. Em contraste com entidades microscópicas como spins ou átomos, eles tendem a ser bem acoplados a outros circuitos, o que os torna atraentes do ponto de vista de leitura e de implementação de portas. Os autores mostram como recentemente novos projetos de qubits supercondutores baseados em circuitos de junção múltipla permitiram o isolamento de perturbações eletromagnéticas extrínsecas indesejadas. Os autores ainda discutem como a

[6] Feynman [7] inclui ainda as portas CCNOT, EXCHANGE E FANOUT, que são derivadas das portas NOT e CNOT.

decoerência do qubit é afetada pelo ruído intrínseco da junção e o que pode ser feito para minorá-lo.

Sob uma perspectiva mais futurista cabe citar a computação quântica topológica, ainda que esta constitua até o presente um campo eminentemente teórico e careça de desenvolvimento experimental. Em um artigo de revisão C. Nayak, S.H. Simon, A. Stern, M. Freedman e S. D. Sarma [38] descrevem a pesquisa atual neste campo focando nos conceitos teóricos gerais das estatísticas não-abelianas em relação à computação quântica topológica, na compreensão de estados $\nu = 5/2$ de Hall quânticos não-abelianos, em experimentos propostos para detectar ânions não-abelianos e sobre arquiteturas propostas para um computador quântico topológico. Tanto os fundamentos matemáticos da computação quântica topológica como a física do tema são abordados, usando o estado quântico fracional de Hall como o arquétipo de um estado topológico não-abeliano que permite a computação quântica tolerante a falhas.

4.2.2
Implementação Física de Portas de um Qubit

Conforme já vimos, o operador rotacional pode ser representado matricialmente como

$$\mathcal{R}_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix} \text{ com } \phi \in \mathbb{R}.$$

Sabemos que $R_{\frac{\pi}{2}}$ e $R_{\frac{\pi}{4}}$, juntamente com as portas de Hadamard e CNOT, constituem um conjunto universal de portas.

Vamos nos limitar aqui à apresentação de um método para realização física do operador $\mathcal{R}_\phi$.[7]

Espelhos, refratores e divisores de feixes de luz são três dos dispositivos mais acessíveis experimentalmente para manipular estados de fótons. Em particular os refratores são usados para a implementação física de operadores rotacionais. Um refrator é uma lâmina de um meio transparente com índice de refração n. A propagação de um feixe de fótons com frequência ω nesse meio por uma distância L muda a fase do fóton por e^{ikL} com $k = \frac{n\omega}{c_0}$ onde c_0 é a velocidade da luz no vácuo.

Assim, se se deseja realizar fisicamente um operador rotacional R_ϕ deve-se inicialmente escolher a frequência ω, o meio e o comprimento L do dispositivo de forma a satisfazer a condição $\phi = \frac{n\omega L}{c_0}$.

Para a realização física do operador rotacional usam-se cristais birrefringentes, que atuam sobre o feixe de fótons incidente de acordo com sua

[7] Para um maior detalhamento, ver Nielsen e Chuang [35], pp. 287-290.

polarização: não ocorrerá mudança de fase do campo elétrico se este oscilar em um certo plano específico e muda-se essa fase por $e^{ikL} = e^{i\phi}$ se o campo elétrico oscilar no plano perpendicular ao primeiro.[8]

No primeiro caso associa-se o campo elétrico do feixe de fótons ao qubit $|0\rangle$ e no segundo caso ao qubit $|1\rangle$.

Assim a lâmina de cristal birrefringente atua sobre o feixe de fótons incidente operando sobre os elementos da base computacional de tal forma que teremos:

$$|0\rangle \longrightarrow |0\rangle$$

$$|1\rangle \longrightarrow e^{i\phi}|1\rangle$$

satisfazendo assim o objetivo de realização física do operador rotacional.

Segundo Nielsen e Chuang [35] p. 203, o modelo baseado em circuitos quânticos é equivalente à máquina de Turing quântica, uma generalização da máquina de Turing clássica. Considerando essa equivalência, veremos no capítulo 7 como os operadores e as portas aqui apresentados serão usados para modelar por meio de circuito a parte central (eminentemente quântica) do algoritmo de fatoração de Shor.

Porém antes disso passaremos a uma descrição completa passo a passo do algoritmo e de suas principais características e propriedades no próximo capítulo.

[8] Obviamente ambos esses planos contêm a reta de propagação do feixe de fótons em questão.

5
O Algoritmo de Fatoração de Shor: Apresentação Passo a Passo e Exemplos

5.1
Introdução: Resumo do Algoritmo e Considerações Iniciais

O objetivo do algoritmo de fatoração de Shor é fatorar um inteiro N ímpar composto, com pelo menos dois fatores primos distintos[1]. Portanto antes de aplicá-lo é necessário:

- testar a paridade de N (se N for par, procede-se a divisões sucessivas por 2 até se obter um número ímpar);

- testar a primalidade de N (se N for primo não tem sentido aplicar o algoritmo);

- testar se N é potência de um primo.

Cabe mencionar que há algoritmos eficientes em computação clássica para teste de primalidade e para fatoração de inteiro potência de um primo.[2]

O algoritmo propriamente dito é constituído de cinco passos, a saber:

Passo 1. Escolha aleatoriamente m inteiro positivo, $1 \leq m \leq N-1$ e:

- se $\text{mdc}(m, N) \neq 1$, obteve-se um fator não trivial de N **FIM**

- se $\text{mdc}(m, N) = 1$, ou seja, se $m \in (\mathbb{Z}/(N))^{\times}$, então deve-se determinar a ordem multiplicativa de m módulo N, aplicando o passo 2 a seguir.

Passo 2. Use computação quântica (e clássica) para determinar o período P da função f dada por

$$f : \mathbb{Z} \to (\mathbb{Z}/(N))^{\times}$$

[1] Este capítulo tem como principal referência Lomonaco [19]. Nele apresentaremos conceitos, definições, propriedades, lemas, teoremas e corolários indispensáveis à compreensão e à avaliação do desempenho do algoritmo, nos abstendo do desenvolvimento das provas dos teoremas.

[2] Cf. Nielsen e Chuang [35], pp. 234, 642, 643.

$$x \mapsto m^x \bmod N$$

Cabe lembrar que $P = \mathrm{ord}_N\, m$ como vimos na seção 2.3 do capítulo 2.

Passo 3. Se P for ímpar vá para o passo 1 e escolha um novo m.

Se P for par vá para o passo 4.

Lembramos que $\mathrm{Prob}(P \text{ ser ímpar} \mid m \in (\mathbb{Z}/(N))^{\times}) \leq \frac{1}{2^k}$, onde k é o número de fatores primos distintos de N, de acordo com o teorema 13 do capítulo 2.

Passo 4. Como $P = \mathrm{ord}_N\, m$ temos:

$$(m^{\frac{P}{2}}-1)(m^{\frac{P}{2}}+1) = m^P - 1 \equiv 0 \pmod N \qquad \text{e} \qquad (m^{\frac{P}{2}}-1) \not\equiv 0 \pmod N$$

Se $(m^{\frac{P}{2}}+1) \equiv 0 \pmod N$, temos a fatoração trivial $N = 1 \cdot N$, devemos ir para o passo 1 e escolher um novo m.

Se $(m^{\frac{P}{2}}+1) \not\equiv 0 \pmod N$, vá para o passo 5.

Sabemos que $\mathrm{Prob}((P \text{ ser par}) \wedge (m^{\frac{P}{2}}+1 \not\equiv 0 \pmod N)) > 1 - \frac{2^{k+1}-1}{2^k(2^k-1)}$ onde $k =$ número de fatores primos distintos de N, conforme o teorema 15 do capítulo 2.

Passo 5. Obtenha $d = \mathrm{mdc}(m^{\frac{P}{2}} - 1, N)$, um fator não trivial de N.

FIM

5.1.1 Considerações sobre o Resumo do Algoritmo

O algoritmo de Shor é essencialmente estocástico. O **passo 2** constitui sua **parte central** (eminentemente mas não exclusivamente quântica) e visa obter o período P de uma função periódica dada. A aplicação de computação quântica nesta parte constitui a contribuição pioneira de Shor.

Devido a sua natureza essencialmente estocástica, o algoritmo está sujeito a ter que retornar ao início após a execução de alguns passos. Isto pode se dar no passo 3 ou no passo 4, conforme vimos acima, mas também pode se dar no passo 2, conforme veremos neste capítulo.

Cabe aqui lembrar que a execução bem sucedida dos passos 3 e 4 depende exclusivamente da escolha aleatória do inteiro m no passo 1 e que a probabilidade de obter sucesso nesses dois passos é estritamente superior a $\frac{5}{12}$, conforme o corolário 8 do capítulo 2.

Resta analisar o comportamento probabilístico do algoritmo ao longo da execução do passo 2. Passamos agora à descrição detalhada desse passo, iniciando pelos procedimentos necessários à sua preparação. Trata-se somente de uma descrição onde serão apresentados alguns conceitos, definições, propriedades e teoremas, nos abstendo de desenvolver demonstrações.

Entretanto, duas das propriedades aqui apresentadas serão reapresentadas no capítulo 6 como teoremas que serão demonstrados, devido a sua relevância central no cálculo da probabilidade de sucesso no passo 2 do algoritmo de Shor.

5.2
A Parte Central do Algoritmo

5.2.1
Preparação da Parte Quântica

Trata-se aqui de preparar os elementos indispensáveis para iniciar a parte quântica do algoritmo, tanto em sentido de reserva de espaço em memória de computador quanto em sentido propriamente quântico de preparação de um experimento, isto é, de preparar estados físicos que serão representados por vetores em espaços de Hilbert. Contudo nesta subseção vamos nos ater apenas à representação vetorial dos estados físicos acrescida de comentários pontuais pertinentes à demanda por espaço de memória.

A preparação se inicia com a identificação dos únicos inteiros Q e L tais que

$$N^2 \leq Q = 2^L < 2N^2$$

Seja $S_Q = \{0, 1, ..., Q-1\}$.

Construa dois registros quânticos $|REG1\rangle$ e $|REG2\rangle$ tensoriados, com L qubits cada um, obtendo:

$$|REG1\rangle|REG2\rangle = |a\rangle|f(a)\rangle = |a\rangle|b\rangle = |a_0a_1...a_{L-1}\rangle|b_0b_1...b_{L-1}\rangle.$$

Lembramos que em computação clássica representa-se um inteiro a, $0 \leq a \leq 2^L - 1$ em base 2 como $a = \sum_{j=0}^{L-1} a_j 2^j$ onde $a_j \in \{0, 1\}$.

A versão do inteiro a em computação quântica é o vetor de estado tensoriado $|a\rangle = |a_0a_1...a_{L-1}\rangle = \otimes_{j=0}^{L-1}|a_j\rangle$ onde $a_j \in \{0, 1\}$.

Assim, enquanto em computação clássica a é um inteiro positivo, em computação quântica $|a\rangle$ é um vetor de estado em um espaço de Hilbert em $\mathbb{C}^{2^L}$.

Como em computação clássica L bits de memória são suficientes para armazenar um número a inteiro com $0 \leq a \leq 2^L - 1$, a condição $N^2 \leq Q = 2^L < 2N^2$ pode parecer exagerada para armazenar classicamente um inteiro N. Entretanto no algoritmo de Shor é necessário reservar em computação quântica 2^L qubits de memória para armazenar cada um dos vetores de estado $|a\rangle$ e $|b\rangle$, pois, como veremos posteriormente, teoremas que garantem propriedades estocásticas essenciais do algoritmo se baseiam nessa condição.

Finalmente, ainda nesta etapa preparatória devemos nos referir à transformada de Fourier discreta (DFT), que será aplicada na parte quântica e contribuirá decisivamente para o breakthrough do algoritmo de Shor em termos de eficiência computacional.

Definição: Transformada de Fourier discreta de Q pontos de um vetor $|y\rangle$ em um espaço vetorial de dimensão Q é o vetor de estado $DFT|y\rangle$ dado por

$$DFT|y\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} w^{xy}|x\rangle,\ w = e^{\frac{2\pi i}{Q}}$$

Na base canônica $|0\rangle, |1\rangle, ..., |Q-1\rangle$ a transformada de Fourier discreta é implementada como uma transformação unitária cujo operador em forma matricial é dado pela matriz $A \in \mathbb{C}^{Q\times Q}$ com

$$A_{xy} = \frac{1}{\sqrt{Q}} w^{xy}$$

5.2.2
Descrição da Parte Central do Algoritmo

Conforme apresentado no passo 2 do resumo, a parte central do algoritmo tem como objetivo determinar o período P da função f dada por

$$f : \mathbb{Z} \longrightarrow (\mathbb{Z}/(N))^{\times}$$

$$x \mapsto m^x \bmod N$$

A obtenção de P, dado m, é constituída por seis passos, passo 2.0 a passo 2.5, dos quais apenas o passo 2.5 não utiliza computação quântica.

Passo 2.0. Inicialize com os registros zerados, isto é, comece preparando *fisicamente* o estado inicial representado pelo vetor

$|\psi_0\rangle = |REG1\rangle|REG2\rangle = |0\rangle|0\rangle = |00...0\rangle|00...0\rangle = (\otimes_{j=0}^{L-1}|0\rangle) \otimes (\otimes_{j=0}^{L-1}|0\rangle)$ [3]

[3]Aqui mais uma vez notamos a representação por produto tensorial. Doravante não

Passo 2.1. Aplique o operador de Hadamard ao $|REG1\rangle$, fixando $|REG2\rangle$=$|0\rangle$=$|00...0\rangle$, obtendo

$$|\psi_1\rangle = \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|0\rangle$$

Notamos que agora $|REG1\rangle$ armazena os estados $|0\rangle$, $|1\rangle$, ..., $|Q-1\rangle$ **em distribuição uniforme (estados equiprováveis) e em superposição**: $|REG1\rangle$ pode ser fatorado no produto tensorial

$$\bigotimes_{x=0}^{Q-1} (\frac{|0\rangle + |1\rangle}{\sqrt{2}})$$

enquanto $|REG2\rangle$ se mantém "zerado" ($|REG2\rangle = |0\rangle$=$|00...0\rangle = \otimes_{j=0}^{L-1}|0\rangle$).

Passo 2.2. Seja U_f a transformação linear unitária tal que $U_f|x\rangle|k\rangle = |x\rangle|k + f(x)\rangle$.

Aplique U_f a $|\psi_1\rangle$ obtendo $|\psi_2\rangle$=$\frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle|f(x)\rangle$. Para o algoritmo de Shor é de interesse apenas o caso particular em que $k = 0$ (pois inicializamos com $|REG2\rangle = 0$) e $f(x) = m^x \bmod N$ (pois o objetivo principal é determinar o período desta função) [4].

Nas palavras de Shor (1997), U_f constitui "the bottleneck of the quantum factoring algotithm". A obtenção de $|\psi_2\rangle$ a partir de $|\psi_1\rangle$ ocorre em nível quântico. A aplicação da transformação linear unitária U_f se dá mediante a *transformação física* do estado representado pelo vetor $|\psi_1\rangle$ no estado representado pelo vetor $|\psi_2\rangle$. Trata-se portanto de um resultado obtido por procedimento físico e não por processamento numérico.

Notamos agora que $|\psi_2\rangle$ é um **emaranhamento** de $|REG1\rangle$ e $|REG2\rangle$ (em geral não pode ser fatorado em produto tensorial).

Passo 2.3. Aplique DFT ao $|REG1\rangle$ obtendo $|\psi_3\rangle$=$\frac{1}{Q} \sum_{y=0}^{Q-1} |y\rangle|\gamma(y)\rangle$
onde $|\gamma(y)\rangle$=$\sum_{x=0}^{Q-1} w^{xy}|f(x)\rangle$

Passo 2.4.
Meça *fisicamente* $|REG1\rangle$ na base computacional, obtendo $|y_0\rangle$,
onde $y_0 \in$ {0,1, ..., Q-1} e Prob(obter y_0) = $\frac{\||\gamma(y_0)\rangle\|^2}{Q^2}$.

insistiremos mais em comentar esta distinção essencial da computação quântica com relação à clássica.

[4]Cabe mencionar que um caso particular da porta U_f é a porta CNOT apresentada na seção 4.1 do capítulo 4.

Após ter realizado esta medida, podemos ignorar os registros $|REG1\rangle$ e $|REG2\rangle$ e notar que criamos uma **distribuição de probabilidades** no espaço amostral $\{0, 1, \ldots, Q-1\}$. Em outras palavras, o propósito de se executar os passos 2.0 a 2.4 foi o de criar (mediante computação quântica) uma **fonte estocástica** finita que gera, mediante tranformação física, um estado representado pelo vetor $|y_0\rangle$ onde $y_0 \in \{0, 1, ..., Q-1\}$ com distribuição de probabilidades conhecida.

O restante do algoritmo tem como objetivo obter o período da função f a partir de inteiros y gerados por essa fonte estocástica, que foi criada para isso. Como veremos de forma detalhada, as propriedades dessa fonte são importantes para uma avaliação do desempenho do algoritmo. Nesse sentido o fato de a distribuição de probabilidades de y não ser uniforme, mas privilegiar valores promissores para a obtenção do período da função f, é fundamental.

Passo 2.5. Obtenha o período P da função f, a partir do valor y_0 obtido no passo 2.4, aplicando o procedimento (clássico) a seguir:

- Expanda a fração $\frac{y_0}{Q}$ em fração contínua, obtendo os convergentes $\frac{p_n}{q_n}$, $n = 1, 2, ..., M$
- para $n = 1$ até $n = M$, teste para verificar se $q_n = P$, calculando

$$m^{q_n} = \prod_{i=1}^{n} (m^{2i})^{q_{n,i}} \pmod{N}$$

  onde $q_n = \sum_{i=0}^{n} q_{n,i} 2^i$ é a expressão binária de q_n.

- Se $m^{q_n} \equiv 1 \pmod{N}$, obtenha o período $P = q_n$ e vá para o passo 3 do algoritmo de Shor, saindo de sua parte central, que foi bem sucedida. Se não, continue o loop (incremente o valor de n).
- Final do loop.
- Se chegar até este ponto e as M tentativas dentro do loop fracassarem [5] volte ao passo 2.0 e refaça a parte central desde o início.[6]

Veremos posteriormente que a probabilidade de fracasso em cada interação do loop acima é menor que 1. Portanto a probabilidade de sucesso cresce a cada interação. Como em geral estamos interessados em fatorar um grande

[5] Lomonaco [19] comenta que, se isso ocorrer, então "you are a very unlucky quantum computer scientist".

[6] Na prática medições destroem o estado. Porém mesmo se fosse possível fazer uma medição ideal (voltando ao passo 2.4), esta segunda medição simplesmente reproduziria o resultado. Assim não tem sentido simplesmente voltar ao passo 2.4 e medir novamente $|REG1\rangle$.

inteiro N, $Q \geq N^2$ e $y \in \{0, 1, ..., Q-1\}$, então a probabilidade de fracasso do algoritmo é muito pequena, para cada valor inicial escolhido para a base m. Por outro lado, considerando-se o algoritmo como um todo, temos em princípio $(N-1)Q \geq (N-1)N^2$ loops independentes (já que $m \in \{1, ..., N-1\}$, cada um deles com probabilidade de sucesso positiva.

A seguir vamos apresentar a distribuição de probabilidades associada à fonte estocástica do algoritmo, que irá respaldar essas afirmações.

5.3
O Cerne da Abordagem Quântica: A Fonte Estocástica do Algoritmo

Nesta seção e também na próxima apresentaremos propriedades, teoremas e corolários relativos à fonte estocástica do algoritmo de Shor indispensáveis à sua compreensão, nos abstendo do desenvolvimento das provas dos teoremas.

As duas propriedades a seguir descrevem de forma completa a distribuição de probabilidades de y. Devido a sua importância na avaliação da eficiência do algoritmo, elas serão reapresentadas na forma de dois teoremas, com as respectivas demonstrações, no capítulo 6.

Propriedade 1. *Sejam y um valor obtenível pela fonte estocástica do algoritmo de Shor e Prob(y) a probabilidade de uma medida a partir dessa fonte resultar em y. Então, se $P \mid Q$, teremos:*

$$\text{Prob}(y) = \begin{cases} 0, & Py \not\equiv 0 \pmod{Q} \\ \frac{1}{P}, & Py \equiv 0 \pmod{Q} \end{cases}$$

Neste caso, y apresenta apenas P valores com probabilidade positiva, com distribuição uniforme de probabilidades, a saber, os valores $0, \frac{Q}{P}, \frac{2Q}{P}, \ldots, Q - \frac{Q}{P}$. Os demais valores de $y \in \{0, 1, \ldots, Q-1\}$ têm probabilidade nula.

Propriedade 2. *Sejam y um valor obtenível pela fonte estocástica do algoritmo de Shor e Prob(y) a probabilidade de uma medida a partir dessa fonte resultar em y. Então, se $P \nmid Q$, teremos:*

$$\text{Prob}(y) = \begin{cases} g_1(r, P, Q, y), & Py \not\equiv 0 \pmod{Q} \\ g_2(r, P, Q), & Py \equiv 0 \pmod{Q} \end{cases}$$

onde

$$g_1(r, P, Q, y) = \frac{r \,\text{sen}^2(\frac{\pi P y}{Q}(\frac{Q_0}{P} + 1)) + (P - r)\,\text{sen}^2(\frac{\pi P y}{Q}\frac{Q_0}{P})}{Q^2 \,\text{sen}^2(\frac{\pi P y}{Q})},$$

$$g_2(r,P,Q) = \frac{r(Q_0+P)^2 + (P-r)Q_0^2}{Q^2P^2}, \quad Q_0 = \left\lfloor \frac{Q}{P} \right\rfloor P, \quad r = Q \bmod P.$$

Neste caso, se Py for múltiplo de Q, Prob(y) independerá de y. Além disso, como veremos posteriormente, $g_2(r,P,Q) > g_1(r,P,Q,y)$ para todo y, indicando assim os valores de y de máxima probabilidade de obtenção (valores "de pico").

Usaremos posteriormente essa propriedade para simular classicamente [7] a aplicação do algoritmo de Shor à fatoração de 35.

Outras propriedades específicas relativas a essa fonte estocástica serão apresentadas na seção 5.4 a seguir e aplicadas tanto na parte principal como na parte final (exclusivamente clássica) do algoritmo (passos 3, 4 e 5).

5.4
Propriedades e Conceitos Aplicados na Avaliação da Probabilidade de Sucesso dos Passos Finais do Algoritmo

A parte final do algoritmo (passos 3, 4 e 5) consiste basicamente em, conhecidos Q=2^L, a base m, o período P e o valor medido de y (y_0), obter um fator não trivial de N. Esse procedimento está respaldado em teoria dos números e no conhecimento de limitantes inferiores positivos para as probabilidades de sucesso nos passos 3 e 4, conforme apresentado a seguir.

Teorema 16. *Sejam y um valor obtenível pela fonte estocástica do algoritmo de Shor e Prob(y) a probabilidade de uma medida a partir dessa fonte resultar em y. Então*

$$\text{Prob}(y) \geq \begin{cases} g_3(P,N), & 0 < |\{Py\}_Q| \leq \frac{P}{2}(1-\frac{1}{N}) \\ g_4(P,N), & \{Py\}_Q = 0 \end{cases}$$

onde

$$g_3(P,N) = \frac{4}{\pi^2}\frac{1}{P}(1-\frac{1}{N})^2 \qquad g_4(P,N) = \frac{1}{P}(1-\frac{1}{N})^2$$

e $\{a\}_Q$ é o resíduo de a de módulo Q de menor magnitude.

Comparando-se com a propriedade 2 acima, notamos que o teorema 16 nos traz informação adicional, qual seja, limitantes inferiores para a probabilidade de ocorrência de valores de y tais que Py seja *próximo* de múltiplo de Q. Estes valores de y situam-se na proximidade dos máximos locais da função de probabilidades de y. Como $g_4(P,N) > g_3(P,N)$ para todo y, aqui notamos a tendência à obtenção mais provável de certos valores específicos de y: os valores de y tais que Py seja *múltiplo* de Q são mais prováveis que valores de y tais que Py seja *próximo* (mas não igual) a um múltiplo de Q.

[7] Mediante uso de planilha em EXCEL.

Usaremos essa propriedade para simular classicamente [8] a aplicação do algoritmo de Shor à fatoração de 35.

Sabemos que $Y = \{y \in \{0, 1, ..., N-1\}; |\{Py\}_Q| \leq \frac{P}{2}\}$ possui P elementos distintos entre si e portanto há uma bijeção entre Y e $S_P = \{0, 1, ..., P-1\}$ e também entre $\{\frac{y}{Q}; y \in Y\}$ e $\{\frac{d}{P}; d \in S_P\}$.

Mais ainda, a função d dada por

$$d : Y \longrightarrow S_P$$

$$y \mapsto \text{round}(\frac{Py}{Q})$$

é uma bijeção com inversa dada por

$$d^{-1} : S_P \longrightarrow Y$$

$$x \mapsto \text{round}(\frac{Qx}{P})$$

Finalmente, notamos que a função bijetora d sugere um caminho para a obtenção de P: partir de $\frac{y}{Q}$ e chegar a $\frac{d(y)}{P}$.

5.5
A Interação entre a Parte Quântica e a Parte Final: Desempenho na Determinação do Período

Voltando à parte quântica do algoritmo, sabemos que, como resultado da medida tomada no passo 2.4, temos um valor inteiro $y_0 \in S_Q = \{1, 2, \ldots, Q-1\}$. Precisamos agora saber como determinar P a partir de y_0, explorando a relação entre os racionais $\frac{y_0}{Q}$ e $\frac{d}{P}$. Veremos que essa busca de P se assemelha a uma "pesca" ou "caçada" em um meio estocástico, com alta probabilidade de sucesso, onde as propriedades da distribuição de probabilidades de y, bem como elementos de teoria dos números, irão desempenhar um papel essencial.

Do teorema 3 do capítulo 2 sabemos que, se c é um real, a, b são inteiros com $b > 0$ e $|c - \frac{a}{b}| \leq \frac{1}{2b^2}$ então $\frac{a}{b}$ é um convergente da expansão em fração contínua de c.

Agora podemos aplicar um corolário útil desse teorema, a saber:

Corolário 9 (condição suficiente para convergência)**.** *Se* $|\{Py\}_Q| \leq \frac{P}{2}$ *então* $\frac{d(y)}{P}$ *é um convergente da expansão em fração contínua de* $\frac{y}{Q}$*.*

Notamos aqui novamente uma propriedade que diz respeito a valores de y tais que Py seja múltiplo ou próximo de múltiplo de Q, isto é, a valores de

[8] Mediante uso de planilha em EXCEL.

y que coincidem ou estão próximos aos valores correspondentes aos picos de Prob(y).

Entretanto, aqui não se trata apenas de, mais uma vez, visualizar limitantes inferiores para as probabilidades de ocorrência de y, mas sim de buscar uma "pista" para a "caçada" do período P. Em uma visão ingênua, poder-se-ia mesmo pensar: "Temos Q e y, logo temos $\frac{y}{Q}$ e o período P seria encontrado imediatamente, pois seria igual ao denominador de um convergente da expansão em fração contínua de $\frac{y}{Q}$".

No entanto, o que o corolário 9 acima garante é apenas que, se for satisfeita certa condição, teremos um n-ésimo convergente da expansão em frações contínuas de $\frac{y}{Q}$, dado por $c_n = \frac{p_n}{q_n}$, onde p_n e q_n são primos entre si e $c_n = \frac{d(y)}{P}$. Mas nada garante que $p_n = d(y)$ e $q_n = P$. Em outras palavras, nada garante o sucesso de nossa "caçada" por P.

Como p_n e q_n são primos entre si, temos que $p_n = d(y)$ e $q_n = P$ se e somente se $d(y)$ e P também forem primos entre si. Por isso a probabilidade de sucesso do algoritmo depende da probabilidade de $d(y)$ e P serem primos entre si.

Contudo temos aqui uma "boa notícia": existe um limitante inferior positivo para a probabilidade de sucesso desta etapa de nossa "caçada", pois sabemos do corolário 5 do capítulo 2 que

$$\frac{\phi(P)}{P} \geq \frac{e^{-\gamma} - \epsilon(P)}{\ln 2} \frac{1}{\log_2 \log_2 N}$$

onde ϕ é a função totiente de Euler, γ é a constante de Euler, $P = \text{ord}_N\, m$ com $1 \leq m \leq N-1$, $\epsilon(P)$ é uma sequência monótona descrescente de números reais positivos que converge para 0 e, além disso, pode ser provado [9] que

Teorema 17. *Prob*$(\text{mdc}(d(y), P) = 1 \mid y \in \{0, 1, ..., N-1\}) \geq \frac{4}{\pi^2} \frac{\phi(P)}{P} (1 - \frac{1}{N})^2$

Logo Prob$(\text{mdc}(d(y), P) = 1 \mid y \in \{0, 1, ..., N-1\})$ tem um limitante inferior positivo: eis a boa notícia.

Recordemos que o passo 2.5 (último passo da parte central do algoritmo) calcula classicamente o período P da função f dada por $x \mapsto m^x \bmod N$, a partir do valor y_0 medido no passo 2.4, mediante o loop:

Para $n = 1$ até que $c_n = 0$:

– Use as relações de recorrência na expansão em frações contínuas para calcular p_n e q_n do n-ésimo convergente $c_n = \frac{p_n}{q_n}$ de $\frac{y_0}{Q}$.

[9] Cf. Lomonaco [19].

– Teste para verificar se $q_n = P$, calculando

$m^{q_n} = \prod_{i=1}^{n} (m^{2i})^{q_{n,i}} \bmod N$
onde $q_n = \sum_{i=0}^{n} q_{n,i} 2^i$ é a expressão binária de q_n

– Se $m^{q_n} \equiv 1 \pmod N$, saia com o valor do período $P = q_n$ e vá para o passo 3 do algoritmo de Shor, saindo da parte central, que se concluiu com sucesso (o período foi obtido!). Se não, continue o loop (incremente o valor de n).

– Final do loop.

– Se chegar até este ponto e todas as tentativas dentro do loop fracassarem[10], volte ao passo 2.0 e refaça a parte quântica desde o início.

Pois bem, já mencionamos que a probabilidade de fracasso em cada interação do loop acima é menor que 1. Agora sabemos que seu complemento, a probabilidade de sucesso, tem um limitante inferior positivo dado por

$$\text{Prob}(\text{mdc}(d(y), P) = 1 \mid y \in \{0, 1, ..., N-1\}) \geq \frac{4}{\pi^2} \frac{\phi(P)}{P} (1 - \frac{1}{N})^2$$

onde ϕ é a função totiente de Euler.

A tabela a seguir ilustra os valores desse limitante, abstraindo o fator $(1 - \frac{1}{N})^2$, que é muito próximo de 1 para N grande.

P	$\phi(P)$	Limitante
2	1	0,2026
3	2	0,2702
4	2	0,2026
5	4	0,3242
6	2	0,1351
7	6	0,3474
8	4	0,2026
9	6	0,2702
10	4	0,1621
11	10	0,3684
12	4	0,1351

[10] Um evento muito improvável: "you are a very unlucky quantum computer scientist!".

Vale lembrar mais uma vez que, em geral, estamos interessados em fatorar um grande inteiro N e adotamos $Q \geq N^2$. Como a medida de y resulta em um entre Q valores inteiros possíveis ($y \in \{0, 1, ..., Q-1\}$), a probabilidade de fracasso do algoritmo é muito pequena para cada valor inicial escolhido para a base m. Como $m \in \{1, ..., N-1\}$, temos em princípio $(N-1)Q \geq (N-1)N^2$ tentativas independentes, cada uma com probabilidade de sucesso positiva, o que resulta em alta probabilidade de sucesso do algoritmo como um todo.

A rigor, somente mediante a execução de todas as $(N-1)Q \geq (N-1)N^2$ tentativas mencionadas se pode assegurar o sucesso do algoritmo com probabilidade igual a 1. Entretanto, o *valor esperado* do número de tentativas necessárias até o término bem sucedido do algoritmo é o inverso de sua probabilidade de sucesso em cada tentativa e é muito inferior a esse número.

5.6
Três Exemplos

Para ilustrar o procedimento vamos fatorar o número 35, o quarto menor inteiro não trivial (ímpar, composto e diferente de potência de um primo), escolhendo três valores para a medida quântica (y_0): o primeiro favorável ao sucesso do algoritmo, os demais apresentando dois tipos de "comportamento" desfavorável.

Devido ao fato de $N = 35$ ser pequeno, podemos obter facilmente o período P da função f definida no passo 2, sem utilizar o passo 2 do algoritmo. Basta calcular $2^x \bmod 35$ para $x = 1, 2, ...$, e guardar como "segredo" o resultado : $P = 12$. Não faremos uso desse "segredo" em nenhum passo da execução do algoritmo nos três exemplos que se seguem, mas exclusivamente nos comentários que encerram cada exemplo.

5.6.1
Um Exemplo de Sucesso

Passo 1: Escolho $m = 2$, que satisfaz $1 \leq m \leq 34$.

Como $\text{mdc}(2, 35) = 1$, sigo para o passo 2, para encontrar o período P da função $f(x) = 2^x \bmod 35$.

Passo 2: Parte central do algoritmo

Preparação da Parte Quântica

Temos $Q = 2^{11} = 2048$ pois $35^2 \leq Q = 2^{11} < 2 \cdot 35^2$.

Passo 2.0. Inicio com

$|\psi_0\rangle = |REG1\rangle|REG2\rangle = |0\rangle|0\rangle = |00...0\rangle|00...0\rangle = (\otimes_{j=0}^{10}|0\rangle) \otimes (\otimes_{j=0}^{10}|0\rangle)$

Notamos aqui que precisamos de 2^{22} qubits para armazenar $|REG1\rangle|REG2\rangle$.

Passo 2.1. Aplico o operador de Hadamard ao $|REG1\rangle$, fixando $|REG2\rangle = |0\rangle = |00...0\rangle$ obtendo

$|\psi_1\rangle = \frac{1}{\sqrt{2048}} \sum_{x=0}^{2047} |x\rangle|0\rangle$

Passo 2.2. Aplicando a transformação linear unitária U_f a $|REG1\rangle|REG2\rangle$ obtenho

$|\psi_2\rangle = \frac{1}{\sqrt{2048}} \sum_{x=0}^{2047} |x\rangle|2^x \bmod 35\rangle$

Passo 2.3. Aplico DFT ao $|REG1\rangle$, obtendo

$|\psi_3\rangle = \frac{1}{\sqrt{2048}} \sum_{y=0}^{2047} |y\rangle|\gamma(y)\rangle$ onde $|\gamma(y)\rangle = \frac{1}{\sqrt{2048}} \sum_{y=0}^{2047} w^{xy}|2^x \bmod 35\rangle$

Passo 2.4. Meço $|REG1\rangle$, obtendo $|y_0\rangle$, por exemplo $|y_0\rangle = |851\rangle$

Passo 2.5. Calculo o período P da função f dada por $x \mapsto 2^x \bmod 35$, a partir do valor $y_0 = 851$ obtido no passo 2.4, aplicando o procedimento a seguir:

– Expando a fração $c = \frac{y_0}{Q} = \frac{851}{2048}$ em fração contínua, obtendo os convergentes $c_n = \frac{p_n}{q_n}$, $n = 1, 2, ..., M$

Aplicando o algoritmo de Euclides e as relações de recorrência vistos na seção 2.5 do capítulo 2, obtenho $a_n, p_n, q_n, c_n, n = 1, 2, ..., M = 7$, conforme a tabela a seguir:

n	0	1	2	3	4	5	6	7
a_n	0	2	2	2	5	1	2	9
p_n	0	1	2	5	27	32	91	851
q_n	1	2	5	12	65	77	219	2048
c_n	0	1/2	2/5	5/12	27/65	32/77	91/219	851/2048

Agora testo, para $n = 1$ até $n = M = 7$ (ou até ter sucesso), para verificar se ocorre $q_n = P$, calculando

$m^{q_n} = \prod_{i=1}^{n} (m^{2i})^{q_{n,i}} \bmod N$ onde $q_n = \sum_{i=0}^{n} q_{n,i} 2^i$ é a expressão binária de q_n.

A tabela acima contém todos os valores da expansão completa de $\frac{y_0}{Q} = \frac{851}{2048}$ em fração contínua. Veremos que somente os valores referentes a n = 0, 1, 2, 3 serão utilizados para obter o período P da função f dada por $x \mapsto 2^x \bmod 35$.

$n = 0$ é trivial e não nos fornece P.

Para $n = 1$, temos $q_1 = 2$ com $2^2 = 4 \not\equiv 1 \pmod{35}$

Para $n = 2$, temos $q_2 = 5$ com $2^5 = 32 \not\equiv 1 \pmod{35}$

Para $n = 3$, temos $q_3 = 12$ com $2^{12} = 4096 \equiv 1 \pmod{35}$

Logo concluímos que $P = 12$ e saímos da parte quântica com sucesso (chegamos ao final do loop), sem precisar usar os valores $n = 4, 5, 6, 7$.

Vou para o passo 3 do algoritmo.

Passo 3: $P = 12$ é par. Sigo para o passo 4.

Passo 4: $2^{\frac{P}{2}} = 2^6 = 64 \not\equiv -1 \pmod{35}$. Vou para o passo 5.

Passo 5: $\text{mdc}(2^{\frac{P}{2}} - 1, 35) = \text{mdc}(63, 35) = 7$

Assim obtemos 7, que é um fator não trivial de 35.

Comentários

Neste exemplo temos $Py_0 = 10212$, que não deve ser considerado próximo de um múltiplo de Q, na perspectiva do teorema 16, pois $|\{Py_0\}_Q| = |\{10212\}_{2048}| = 28 > \frac{P}{2}(1 - \frac{1}{N})$ e temos da propriedade 2 da seção 5.3 que $\text{Prob}(y = 851) = 0,00116353 < \frac{4}{\pi^2}\frac{1}{P}(1 - \frac{1}{N})^2$, um valor compatível com o enunciado do teorema.

Por outro lado $Py_0 = 10212$ também não deve ser considerado suficientemente próximo de um múltiplo de Q segundo o corolário 9, pois $|\{Py_0\}_Q| = |\{10212\}_{2048}| = 28 > \frac{P}{2}$. Portanto não podemos *assegurar* que $\frac{d}{P}$ seja um convergente da expansão em fração contínua de $c = \frac{y_0}{Q} = \frac{851}{2048}$. Todavia verificamos que $d = d(y_0) = d(851) = \text{round}(\frac{12 \cdot 851}{2048}) = 5$ e estamos com sorte, pois $\text{mdc}(d, P) = \text{mdc}(5, 12) = 1$, com $\text{Prob}(\text{mdc}(d, P) = 1) > 0,127$ segundo o teorema 17 ou mais precisamente $\text{Prob}(\text{mdc}(d, P) = 1) = 0,33306337$, conforme pode ser obtido a partir da propriedade 2 da seção 5.3.

Além disso verificamos também (por expansão em fração contínua) que $\frac{d}{P} = \frac{5}{12}$ é um convergente da expansão em fração contínua de $\frac{y_0}{Q} = \frac{851}{2048}$. Com efeito, o corolário 9 expressa apenas um condição *suficiente* para que esse fato ocorra.

5.6.2
Um Exemplo de Fracasso

Neste exemplo, todos os passos até o passo 2.3 são idênticos aos do exemplo anterior.

Passo 2.4.
Meço $|REG1\rangle$, obtendo $|y_0\rangle$, por exemplo $|y_0\rangle = |1367\rangle$.

Passo 2.5. Calculo o período P da função f dada por $x \mapsto 2^x \bmod 35$, a partir do valor $y_0 = 1367$ obtido no passo 2.4, aplicando o procedimento a seguir:

– Expando a fração $c = \frac{y_0}{Q} = \frac{1367}{2048}$ em fração contínua, obtendo os convergentes $c_n = \frac{p_n}{q_n}, n = 1, 2, ..., M$

Aplicando o algoritmo de Euclides e as relações de recorrência vistos na seção 2.5 do capítulo 2, obtemos $a_n, p_n, q_n, c_n, n = 1, 2, ..., M = 4$, conforme a tabela a seguir:

N	0	1	2	3	4
a_n	0	1	2	136	5
p_n	0	1	2	273	1367
q_n	1	1	3	409	2048
c_n	0	1	2/3	273/409	1367/2048

Agora testo, para $n = 1$ até $n = M = 4$ (ou até ter sucesso), para verificar se ocorre $q_n = P$, calculando

$m^{q_n} = \prod_{i=1}^{n} (m^{2i})^{q_{n,i}} \bmod N$ onde $q_n = \sum_{i=0}^{n} q_{n,i} 2^i$ é a expressão binária de q_n.

A tabela acima contém todos os valores da expansão completa de $\frac{y_0}{Q} = \frac{1367}{2048}$ em fração contínua.

$n = 0$ e $n = 1$ são triviais e não nos fornecem P.

Para $n = 2$, temos $q_2 = 3$ com $2^3 = 8 \not\equiv 1 \pmod{35}$

Para $n = 3$ e $n = 4$ temos $q_3 = 409$ e $q_4 = 2408$, que não podem ser iguais a P, pois necessariamente $P < 35$

Esgotamos todos os resultados da expansão em fração contínua de $\frac{y_0}{Q} = \frac{1367}{2048}$ sem obter P e chegamos ao final do loop sem sucesso.

Volto ao passo 2.0 e refaço a parte quântica desde o início.

Comentários

Neste exemplo temos $Py_0 = 16404$, que não é próximo de um múltiplo de Q na perspectiva do teorema 16, pois $|\{Py_0\}_Q| = |\{16404\}_{2048}| = 20 > \frac{P}{2}(1 - \frac{1}{N})$ e temos da propriedade 2 da seção 5.3 que $\text{Prob}(y = 1367) = 0,00228012 < \frac{4}{\pi^2}\frac{1}{P}(1 - \frac{1}{N})^2$, um resultado compatível com o enunciado do teorema.

Verificamos também que $Py_0 = 16404$ não é suficientemente próximo de um múltiplo de Q na perspectiva do corolário 9, pois $|\{Py_0\}_Q| = |\{16404\}_{2048}| = 20 > \frac{P}{2}$. Assim, não se pode assegurar que $\frac{d}{P}$ seja um convergente $c_n = \frac{p_n}{q_n}$ da expansão em fração contínua de $c = \frac{y_0}{Q} = \frac{1367}{2048}$.

Por outro lado, temos $d = d(y_0) = d(1367) = \text{round}(\frac{12 \cdot 1367}{2048}) = 8$ e estamos sem sorte, pois $\text{mdc}(d, P) = \text{mdc}(8, 12) \neq 1$, com $\text{Prob}(\text{mdc}(d, P) = 1) > 0,127$,

segundo o teorema 17, ou mais precisamente $\text{Prob}(\text{mdc}(d, P) = 1) = 0,33306337$, conforme pode ser obtido a partir da propriedade 2 da seção 5.3.

Assim, ainda que $\frac{d}{P}$ fosse igual a um convergente $c_n = \frac{p_n}{q_n}$ da expansão em fração contínua de $\frac{y_0}{Q}$, mesmo assim não teríamos $d = p_n$ nem $P = q_n$.

Finalmente, verificamos que $d(y_0) = \text{round}(\frac{Py_0}{Q}) = \text{round}(\frac{16404}{2048}) = 8$ e $\frac{d}{P} = \frac{8}{12} = \frac{3}{4}$ não é sequer um convergente da expansão em fração contínua de $\frac{y_0}{Q} = \frac{1367}{2048}$ (os convergentes c_0, c_1, c_2, c_3, c_4 são respectivamente 0, 1, $\frac{2}{3}$, $\frac{273}{409}$, $\frac{1367}{2048}$).

5.6.3
Segundo Exemplo de Fracasso

Neste exemplo, todos os passos até o passo 2.3 são idênticos aos do exemplo do item VII.1.

Passo 2.4.
Meço $|REG1\rangle$, obtendo $|y_0\rangle$, por exemplo $|y_0\rangle = |1536\rangle$

Passo 2.5. Calculo o período P da função f dada por $x \mapsto 2^x \bmod 35$, a partir do valor $y_0 = 1536$ obtido no passo 2.4, aplicando o procedimento a seguir:

– Expando a $\frac{y_0}{Q} = \frac{1536}{2048}$ em fração contínua, obtendo os convergentes $c_n = \frac{p_n}{q_n}$, n = 1, 2, ..., M

Aplicando o algoritmo de Euclides e as relações de recorrência, obtenho $a_n, p_n, q_n, , c_n, n = 1, 2, ..., M$ (ver item IV.1), conforme a tabela a seguir:

N	0	1	2
a_n	0	1	3
p_n	0	1	3
q_n	1	1	4
c_n	0	1	3/4

Agora testo, para $n = 1$ até $n = M = 2$ (ou até ter sucesso), para verificar se ocorre $q_n = P$, calculando

$m^{q_n} = \prod_{i=1}^{n} (m^{2i})^{q_{n,i}} \bmod N$ onde $q_n = \sum_{i=0}^{n} q_{n,i} 2^i$ é a expressão binária de q_n.

A tabela acima contém todos os valores da expansão completa de $\frac{y_0}{Q} = \frac{1536}{2048}$ em fração contínua.

$n = 0$ e $n = 1$ são triviais e não nos fornecem P.

Para $n = 2$ temos $q_2 = 4$ com $2^4 = 16 \not\equiv 1 \pmod{35}$

Esgotamos todos os resultados da expansão em fração contínua de $\frac{y_0}{Q} = \frac{1536}{2048}$ sem obter P e chegamos ao final do loop sem sucesso.

Volto ao passo 2.0 e refaço a parte quântica desde o início.

Comentários

Neste exemplo temos $Py_0 = 18432 = 9Q$ e temos da propriedade 2 da seção 5.3 que $\text{Prob}(y = 1536) = 0,08333397 = \frac{1}{P}$ (valor de pico da distribuição de probabilidades de y).

Temos também $d = d(y_0) = d(1536) = \text{round}(12 \cdot \frac{1536}{2048}) = \text{round}(9) = 9$ mas estamos sem sorte, pois $\text{mdc}(d, P) = \text{mdc}(9, 12) \neq 1$, com $\text{Prob}(\text{mdc}(d, P) = 1) > 0,127$, segundo o teorema 17, ou mais precisamente $\text{Prob}(\text{mdc}(d, P) = 1) = 0,33306337$, conforme pode ser obtido a partir da propriedade 2 da seção 5.3.

Com efeito, temos $\frac{d}{P} = \frac{9}{12} = \frac{3}{4}$, a expansão em fração contínua de $\frac{y_0}{Q}$ tem como convergente $c_2 = \frac{p_2}{q_2} = \frac{3}{4}$ mas não é correto que $P = q_2 = 4$.

Temos $\{Py_0\}_Q = \{18432\}_{2048} = 0 < \frac{P}{2}$ e $\frac{d}{P}$ é seguramente um convergente da expansão em fração contínua de $\frac{y_0}{Q} = \frac{1536}{2048}$, já que a condição suficiente para isso é satisfeita (cf. corolário 9).

Ainda assim fracassamos, pois o denominador de $\frac{d}{P}$ tal como o algoritmo o "enxerga" ($q_2 = 4$) não é igual a P, mas apenas um seu divisor (o que é não somente possível, mas necessário, já que $\text{mdc}(d, P) \neq 1$).

Considerando todos os três exemplos apresentados nesta seção, vemos que a escolha inicial de $m = 2$ como base para fatorar $N = 35$ levou ao seguinte cenário em termos de probabilidade de sucesso do algoritmo:

Em primeiro lugar, a condição (desejável) dada por

$$(\text{ord}_N\, m \text{ ser par}) \wedge (m^{\frac{\text{ord}_N\, m}{2}} + 1 \not\equiv 0 \pmod N)$$

foi satisfeita pela escolha aleatória (e feliz) de m. Com efeito, temos

$$\text{ord}_N\, m = P = 12 \qquad \text{e} \qquad m^{\frac{\text{ord}_N\, m}{2}} + 1 = -5 \not\equiv 0 \pmod N.$$

Assim, logo no início do algoritmo (passo 1), tivemos a sorte de evitar dois tipos de fracasso em sua execução.

A propósito, já sabemos do corolário 8 do capítulo 6 que a probabilidade de se ter esse tipo de sorte é superior a $\frac{5}{12}$.

Já a satisfação da condição (também desejável) $\text{mdc}(d, P) = 1$ depende de um valor que resulta de uma medida física (y) e a probabilidade de esta condição ser satisfeita no contexto em pauta (isto é, com $N = 35$ e $m = 2$) é $0,33306337$, muito próxima de $\frac{1}{3}$.

Portanto, em sentido probabilístico, fatorar $N = 35$ aplicando o algoritmo de Shor e usando a base $m = 2$ equivale a lançar um dado e apostar, por exemplo, somente nos resultados 1 e 2, mas sempre podendo repetir o experimento até ocorrer o primeiro sucesso! Nesse contexto, como a probabilidade

de sucesso em cada tentativa é praticamente igual a $\frac{1}{3}$, o número esperado de tentativas até finalizar a execução bem sucedida do algoritmo é de apenas 3!

Uma vez concluída esta exposição do algoritmo, cabe agora voltar nossa atenção com mais profundidade para as propriedades centrais de sua fonte estocástica, expressas como propriedades 1 e 2 neste capítulo, ou seja, para a distribuição de probabilidades dos valores da medida quântica y. O desempenho do algoritmo depende essencialmente dessa distribuição e por esse motivo optamos por dedicar um capítulo à parte para apresentá-las na forma de teoremas com as respectivas demonstrações.

6
Demonstração de Teoremas Relativos à Fonte Estocástica do Algoritmo de Shor

O algoritmo de Shor é essencialmente estocástico e sua eficiência baseia-se essencialmente na probabilidade de sucesso de sua parte quântica. Por essa razão a distribuição de probabilidades dos valores da fonte estocástica utilizada nessa parte é peça chave tanto na compreensão como na aplicação do algoritmo.

Este breve capítulo tem como objetivo apresentar a demonstração dos dois teoremas relativos a tal distribuição, enunciados no capítulo anterior.

Teorema 18. *Sejam y um valor obtenível pela fonte estocástica do algoritmo de Shor e Prob(y) a probabilidade de uma medida a partir dessa fonte resultar em y. Então, se $P \mid Q$, teremos:*

$$\text{Prob}(y) = \begin{cases} 0, & Py \not\equiv 0 \pmod{Q} \\ \frac{1}{P}, & Py \equiv 0 \pmod{Q} \end{cases}$$

Demonstração.

Dos passos 2.3 e 2.4 do algoritmo de Shor temos

$$\text{Prob(obter } y) = \frac{\||\gamma(y)\rangle\|^2}{Q^2} \quad \text{onde} \quad |\gamma(y)\rangle = \sum_{x=0}^{Q-1} w^{xy} |f(x)\rangle \quad , \qquad w = e^{\frac{2\pi i}{Q}}$$

Sejam $S_Q = \{0, 1, \ldots, Q-1\} \subset \mathbb{N}$ e f dada por

$$f : S_Q \to S_Q$$

$$x \mapsto m^x \bmod N$$

A notação acima para o domínio de f remonta à ideia de fonte estocástica, ou melhor, de espaço amostral (sample space), no qual se realiza o experimento

aleatório de escolha de um inteiro y com $0 \leq y \leq Q-1$. S_Q se distingue de $(\mathbb{Z}/(Q))$, um conjunto de classes de equivalência, e lhe é preferível neste contexto, já que $m^{\bar{x}}$, onde $\bar{x}$ é uma classe de equivalência, não faria sentido.

Sabemos que a extensão de f a $\mathbb{Z}$ é periódica com período $P = \mathrm{ord}_N\, m$.

Como $P \mid Q$, tomemos k inteiro tal que $Q = kP$.

Temos então

$$|\gamma(y)\rangle = \sum_{x=0}^{Q-1} w^{xy}|f(x)\rangle = \sum_{x_0=0}^{P-1}\sum_{x_1=0}^{k-1} w^{(Px_1+x_0)y}|f(Px_1+x_0)\rangle$$

$$|\gamma(y)\rangle = \sum_{x_0=0}^{P-1} w^{x_0 y}\left(\sum_{x_1=0}^{k-1} w^{Pyx_1}\right)|f(x_0)\rangle$$

Seja $S_P = \{0, 1, \ldots, P-1\}$.

Como f tem período P, a restrição de f a S_P é injetora e portanto $|f(0)\rangle, |f(1)\rangle, \ldots, |f(P-1)\rangle$ são vetores distintos e ortonormais.

Analisemos agora os dois casos: $Py \equiv 0 \pmod{Q}$ e $Py \not\equiv 0 \pmod{Q}$, lembrando que $w = e^{\frac{2\pi i}{Q}}$.

Se $Py \equiv 0 \pmod{Q}$ temos (para todo x_1 inteiro) $w^{Pyx_1} = 1$ e portanto

$$\langle\gamma(y)|\gamma(y)\rangle = Pk^2, \qquad \mathrm{Prob}(y) = \frac{\langle\gamma(y)|\gamma(y)\rangle}{Q^2} = \frac{1}{P}.$$

Se $Py \not\equiv 0 \pmod{Q}$, somamos a série geométrica e obtemos

$$\sum_{x_1=0}^{k-1} w^{Pyx_1} = \frac{w^{Py(k-1)}w^{Py}-1}{w^{Py}-1} = \frac{w^{Pyk}-1}{w^{Py}-1} = \frac{w^{Qy}-1}{w^{Py}-1}$$

Notamos que $w^{Qy} = 1$ para todo y inteiro e $w^{Py} \neq 1$ para $Py \not\equiv 0 \pmod{Q}$.

Portanto, para $Py \not\equiv 0 \pmod{Q}$, temos $\sum_{x_1=0}^{k-1} w^{Pyx_1} = 0, \langle\gamma(y)|\gamma(y)\rangle = 0$ e $\mathrm{Prob}(y) = 0$.

E o teorema está demonstrado. □

Teorema 19. *Sejam y um valor obtenível pela fonte estocástica do algoritmo de Shor e Prob(y) a probabilidade de uma medida a partir dessa fonte resultar em y. Então, se $P \nmid Q$, teremos:*

$$\mathrm{Prob}(y) = \begin{cases} g_1(r,P,Q,y), & Py \not\equiv 0 \pmod{Q} \\ g_2(r,P,Q), & Py \equiv 0 \pmod{Q} \end{cases}$$

onde

$$g_1(r,P,Q,y) = \frac{r\,\mathrm{sen}^2(\frac{\pi Py}{Q}(\frac{Q_0}{P}+1)) + (P-r)\,\mathrm{sen}^2(\frac{\pi Py}{Q}\frac{Q_0}{P})}{Q^2\,\mathrm{sen}^2(\frac{\pi Py}{Q})},$$

$$g_2(r,P,Q) = \frac{r(Q_0+P)^2 + (P-r)Q_0^2}{Q^2P^2}, \quad Q_0 = \left\lfloor \frac{Q}{P} \right\rfloor P, \quad r = Q \bmod P.$$

Demonstração. Como na demonstração do teorema 1, temos:

$$\text{Prob(obter } y) = \frac{\||\gamma(y)\rangle\|^2}{Q^2} \quad \text{onde} \quad |\gamma(y)\rangle = \sum_{x=0}^{Q-1} w^{xy}|f(x)\rangle \quad , \quad w = e^{\frac{2\pi i}{Q}}.$$

Também adotamos $S_Q = \{0,1,\ldots,Q-1\} \subset \mathbb{N}$ e f dada por

$$f : S_Q \to S_Q$$

$$x \mapsto m^x \bmod N,$$

lembrando novamente que a extensão de f a $\mathbb{Z}$ é periódica com período $P = \mathrm{ord}_N\, m$.

Como $P \nmid Q$, tomemos k e r como os únicos inteiros tais que $Q = Pk + r, k \geq 0, 0 < r < P$.

Notamos que $Q_0 = \left\lfloor \frac{Q}{P} \right\rfloor P = kP$ e temos então

$$|\gamma(y)\rangle = \sum_{x=0}^{Q-1} w^{xy}|f(x)\rangle = \sum_{x=0}^{Q_0-1} w^{xy}|f(x)\rangle + \sum_{x=Q_0}^{Q-1} w^{xy}|f(x)\rangle =$$

$$= \sum_{x_0=0}^{P-1}\sum_{x_1=0}^{k-1} w^{(Px_1+x_0)y}|f(Px_1+x_0)\rangle + \sum_{x_0=0}^{r-1} w^{(Q_0+x_0)y}|f(Q_0+x_0)\rangle =$$

$$= \sum_{x_0=0}^{P-1} w^{x_0y}\left(\sum_{x_1=0}^{k-1} w^{Pyx_1}\right)|f(x_0)\rangle + \sum_{x_0=0}^{r-1} w^{x_0y}w^{Q_0y}|f(x_0)\rangle =$$

$$= \left(\sum_{x_1=0}^{k-1} w^{Pyx_1}\right)\sum_{x_0=0}^{P-1} w^{x_0y}|f(x_0)\rangle + w^{Q_0y}\sum_{x_0=0}^{r-1} w^{x_0y}|f(x_0)\rangle =$$

$$= \left(\sum_{x_1=0}^{k-1} w^{Pyx_1}\right)\sum_{x_0=0}^{r-1} w^{x_0y}|f(x_0)\rangle + \left(\sum_{x_1=r}^{k-1} w^{Pyx_1}\right)\sum_{x_0=r}^{P-1} w^{x_0y}|f(x_0)\rangle +$$

$$+w^{Pky}\sum_{x_0=0}^{r-1} w^{x_0 y}|f(x_0)\rangle =$$

$$= \left(\sum_{x_1=0}^{k} w^{Pyx_1}\right)\sum_{x_0=0}^{r-1} w^{x_0 y}|f(x_0)\rangle + \left(\sum_{x_1=r}^{k-1} w^{Pyx_1}\right)\sum_{x_0=r}^{P-1} w^{x_0 y}|f(x_0)\rangle.$$

Como f tem período P, a restrição de f a S_P é injetora e portanto $|f(0)\rangle, |f(1)\rangle, \ldots, |f(P-1)\rangle$ são vetores distintos e ortonormais.

Temos então

$$\langle\gamma(y)|\gamma(y)\rangle = r\left|\sum_{x_1=0}^{k} w^{Pyx_1}\right|^2 + (P-r)\left|\sum_{x_1=0}^{k-1} w^{Pyx_1}\right|^2.$$

Analisemos agora os dois casos: $Py \equiv 0 \pmod{Q}$ e $Py \not\equiv 0 \pmod{Q}$, lembrando que $w = e^{\frac{2\pi i}{Q}}$.

Se $Py \equiv 0 \pmod{Q}$, temos (para todo x_1 inteiro) $w^{Pyx_1} = 1$, e portanto

$$\langle\gamma(y)|\gamma(y)\rangle = r(k+1)^2 + (P-r)k^2 = \frac{r(Q_0+P)^2 + (P-r)Q_0^2}{P^2}$$

e teremos

$$\text{Prob}(y) = \frac{\langle\gamma(y)|\gamma(y)\rangle}{Q^2} = \frac{r(Q_0+P)^2 + (P-r)Q_0^2}{Q^2P^2}.$$

Se $Py \not\equiv 0 \pmod{Q}$, somamos as séries geométricas e obtemos

$$\langle\gamma(y)|\gamma(y)\rangle = r\left|\frac{w^{Py(k+1)}-1}{w^{Py}-1}\right|^2 + (P-r)\left|\frac{w^{Pyk}-1}{w^{Py}-1}\right|^2.$$

Aplicando a identidade $\left|e^{i\theta}-1\right|^2 = 4\sin^2\left(\frac{\theta}{2}\right)$ obtemos

$$\text{Prob}(y) = \frac{\langle\gamma(y)|\gamma(y)\rangle}{Q^2} = \frac{r\,\text{sen}^2(\frac{\pi Py}{Q}(\frac{Q_0}{P}+1)) + (P-r)\,\text{sen}^2(\frac{\pi Py}{Q}\frac{Q_0}{P})}{Q^2\,\text{sen}^2(\frac{\pi Py}{Q})}.$$

Fica assim concluída a demonstração. □

No capítulo 3 apresentamos um esboço de um computador quântico e, no capítulo 4, as principais portas a serem utilizadas em uma representação adequada da parte quântica do algoritmo de Shor por meio de circuito. Cabe agora construir essa representação usando portas devidamente especificadas e articuladas em um circuito completo.

7
A Parte Quântica do Algoritmo de Shor como Circuito

7.1
Introdução

A representação mais aceita de um computador clássico é a máquina universal de Turing. Contudo quando se trata de computação quântica a representação através de circuito pode ser mais adequada para a visualização do que sucede passo a passo ao se executar o algoritmo.

Vamos agora aproveitar o esboço de computador quântico apresentado no capítulo 3 e as principais portas quânticas apresentadas no capítulo 4 para construir uma representação da parte quântica do algoritmo em forma de circuito.

Lembramos inicialmente que a parte central (passo 2) do algoritmo de fatoração de Shor aplica computação quântica (e clássica) para determinar o período P da função f dada por

$$f : \mathbb{Z} \to (\mathbb{Z}/(N))^{\times}$$

$$x \mapsto m^x \bmod N$$

Nessa parte central, a computação quântica é aplicada nos passos 2.0 a 2.4, que agora passamos a representar na forma do circuito da Figura 7.1. Assim a aplicação de computação quântica no algoritmo pode ser visualizada e melhor compreendida em forma de "hardware".

Na Figura 7.1, o desenvolvimento sucessivo dos cinco passos da parte quântica do algoritmo se dá de forma sequencial, da esquerda para direita.

Assim os vetores de estado $|\psi_0\rangle, |\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle$ e $|\psi_4\rangle$ representam as saídas dos passos 2.0, 2.1, 2.2, 2.3 e 2.4, respectivamente.

Já o módulo DFT representa o operador transformada de Fourier discreta aplicado no passo 2.3 enquanto o módulo U_f representa o operador linear unitário aplicado no passo 2.2 dado por $U_f(|j\rangle|k\rangle) = |j\rangle|k + x^j\rangle$, onde a soma

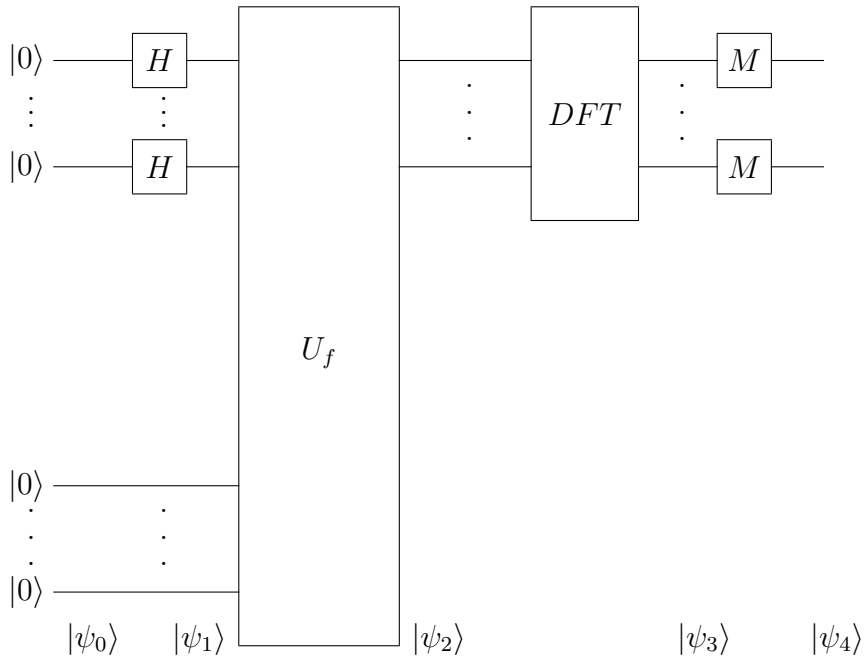

Figura 7.1: Circuito básico da parte quântica

é módulo N e $|j\rangle$ e $|k\rangle$ são os estados do primeiro e do segundo registros, respectivamente. Os módulos H são portas de Hadamard e os módulos M são medidores.

Para que não fiquem apenas representados como meras "caixas pretas", o módulo DFT e o módulo U_f devem ser decompostos "a nível de hardware" em termos de portas universais.

Conforme mencionamos na seção 5.2.2 do capítulo 5, nas próprias palavras de Shor (1997) U_f constitui "the bottleneck of the quantum factoring algorithm". Neste capítulo nos limitaremos a decompor a transformada de Fourier discreta (DFT) em termos de portas universais.

7.2 Operador de Hadamard e Transformada de Fourier: Preparação para Representação como Portas em Circuitos

7.2.1 O Operador de Hadamard

Na parte quântica do algoritmo de Shor destaca-se a aplicação de dois operadores lineares unitários: o operador de Hadamard no passo 2.1 e a transformada de Fourier no passo 2.3.

A porta de Hadamard com n qubits $|\phi_1\rangle, |\phi_2\rangle, ...|\phi_n\rangle$, tem como output o produto tensorial

$$H|\phi_1\rangle \otimes H|\phi_2\rangle \otimes ... \otimes H|\phi_n\rangle$$

Já vimos no capítulo 5 que no caso de uma porta com n qubits "zerados" no input a aplicação do produto tensorial de Hadamard tem como output:

$$H^{\otimes n}|0\rangle^{\otimes n} = H^{\otimes n}|0,...,0\rangle = (H|0\rangle)^{\otimes n} = (\frac{|0\rangle + |1\rangle}{\sqrt{2}})^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$$

Assim o produto tensorial de n operadores de Hadamard produz uma superposição de todos os estados da base computacional com igual amplitude (igual peso), quando aplicado ao estado $|0\rangle^{\otimes n}$ com n qubits. Esta superposição é o output do passo 2.1 da parte quântica, representado por $|\psi_1\rangle$ na figura 7.1.

7.2.2
A Transformada de Fourier

O cálculo eficiente da transformada de Fourier é crucial para o desempenho do algoritmo de Shor e decisivo para caracterizar seu breakthrough em comparação com os algoritmos clássicos conhecidos de fatoração de inteiros.

Inicialmente devemos recordar que a transformada de Fourier discreta de Q pontos de um vetor $|j\rangle$ da base computacional de um espaço vetorial de dimensão Q é o vetor de estado $DFT|j\rangle$ dado por

$$DFT|j\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} w^{kj}|k\rangle$$

onde

$$w = e^{\frac{2\pi i}{Q}}$$

No algoritmo de Shor, para armazenar o inteiro N a ser fatorado, usamos n bits de forma que

$$2^{n-1} < N \leq 2^n = Q$$

Notamos que na expressão para $DFT|j\rangle$ acima o lado direito tem Q termos e que a base computacional tem Q estados. Logo a complexidade para calcular classicamente a transformada de Fourier da base computacional é $O(Q^2) = O(2^{2n})$, conforme a notação assintótica introduzida na seção 2.11 do capítulo 2. Com o desenvolvimento da transformada de Fourier clássica rápida, essa complexidade foi reduzida para $O(n2^n)$ e mesmo assim continua a crescer exponencialmente com n.

Com a aplicação de computação quântica essa complexidade pode ser drasticamente reduzida para complexidade polinomial mediante recurso a

paralelismo, conforme veremos a seguir.

Inicialmente $DFT|j\rangle$ deve ser fatorado. O primeiro passo é escrever k em base 2:

$$k = \sum_{l=1}^{n} k_l 2^{n-l}$$

Obtemos

$$DFT|j\rangle = \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} w^{kj}|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} ... \sum_{k_n=0}^{1} e^{2\pi ij \sum_{l=1}^{n} \frac{k_l}{2^l}} |k_1\rangle \otimes ... \otimes |k_n\rangle$$

$$DFT|j\rangle = \frac{1}{\sqrt{2^n}} \sum_{k_1=0}^{1} ... \sum_{k_n=0}^{1} \bigotimes_{l=1}^{n} e^{2\pi ij \frac{k_l}{2^l}} |k_l\rangle$$

Intercambiando somas e produtos, temos:

$$DFT|j\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} \sum_{k_l=0}^{1} e^{2\pi ij \frac{k_l}{2^l}} |k_l\rangle$$

Finalmente obtemos:

$$DFT|j\rangle = \frac{1}{\sqrt{2^n}} \bigotimes_{l=1}^{n} (|0\rangle + e^{\frac{2\pi ij}{2^l}}|1\rangle)$$

$$DFT|j\rangle = \frac{|0\rangle + e^{\frac{2\pi ij}{2}}|1\rangle}{\sqrt{2}} \otimes \frac{|0\rangle + e^{\frac{2\pi ij}{2^2}}|1\rangle}{\sqrt{2}} \otimes ... \otimes \frac{|0\rangle + e^{\frac{2\pi ij}{2^n}}|1\rangle}{\sqrt{2}}$$

Com esta fatoração chegamos a um ponto crucial: a complexidade para o cálculo de cada $DFT|j\rangle$ passa a ser $O(n)$ (complexidade linear), pois há n termos no produto. Como a base computacional é constituída por n elementos, esse procedimento deverá se repetir n vezes e assim a aplicação do operador DFT quântico terá complexidade $O(n^2)$.

Recordemos que a complexidade da transformada de Fourier rápida clásica é $O(n2^n)$ (complexidade exponencial), já que o cálculo se realiza sobre cada um dos 2^n elementos da base computacional, um de cada vez.

Aqui se pode identificar o breakthrough introduzido pelo algoritmo de fatoração de Shor: trata-se do primeiro algoritmo quântico a introduzir uma redução exponencial no tempo de processamento na resolução de um problema matemático, comparativamente aos algoritmos conhecidos de computação clássica, acenando assim para a possibilidade da existência de outros algoritmos quânticos com essa mesma propriedade.

Como se explica esse breakthrough? Em computação quântica aplica-se

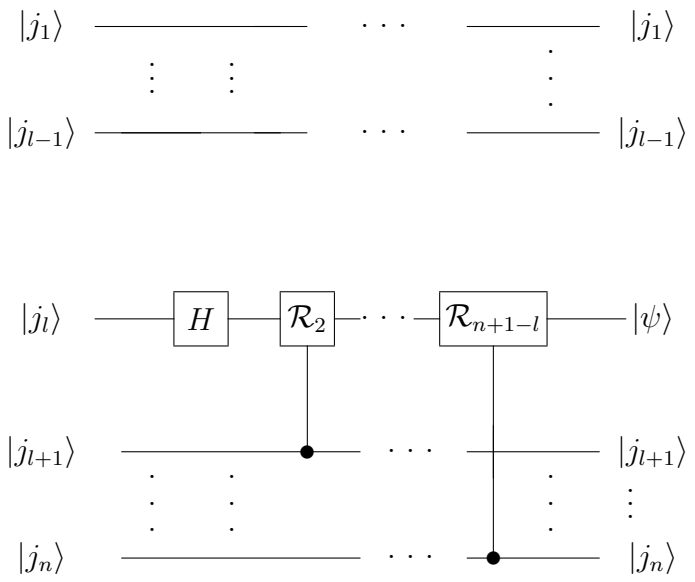

Figura 7.2: Circuito com portas $\mathcal{R}$ e de Hadamard

o paralelismo quântico e a transformada de Fourier de um estado qualquer

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

o qual tem um número exponencial de termos, é calculada com uma única aplicação da transformada de Fourier quântica. A transformada de Fourier dos 2^n elementos da base computacional é realizada simultaneamente ("paralelamente") e portanto a complexidade da transformada de Fourier quântica é medida pelo "tamanho" do circuito (n).

Assim, a expressão fatorada para $DFT|j\rangle$ acima, eminentemente quântica, é a chave da "desaceleração exponencial" propiciada pelo algoritmo de Shor. E é também a chave para a representação de sua parte quântica por meio de circuito, isto é, em termos de portas universais, como veremos a seguir.

7.3 O Circuito: Considerações sobre Complexidade

Consideremos o circuito da Figura 7.2. Nela, o valor dos qubits $|j_m\rangle, m \neq l$, não muda.

Passemos ao qubit $|j_l\rangle$. Este deve mudar sob a ação dos operadores H (porta de Hadamard), $\mathcal{R}_2, \mathcal{R}_3, ..., \mathcal{R}_{n+1-l}$, onde $\mathcal{R}_k, k = 2, 3, ..., n+1-l$ é o

operador rotacional representado pela matriz unitária

$$\mathcal{R}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{pmatrix}$$

Assim cada porta (operador) $\mathcal{R}_k$ é controlada pelo qubit $|j_{k+l-1}\rangle$, de forma que se $\mathrm{j}_{k+l-1} = 0$, $\mathcal{R}_k$ é substituído pelo operador identidade (ausência da ação) e se $\mathrm{j}_{k+l-1} = 1$, $\mathcal{R}_k$ entra em ação. Tendo em vista a efetiva construção do circuito, é importante notar que as portas controladas podem ser implementadas através do uso de portas de um qubit e da porta $CNOT$ (cf. Nielsen e Chuang [35], pp. 178-185).

Notamos que para efeito de cálculo cada porta de dois qubits $\mathcal{R}_k$ controlada pelo qubit $|j_{k+l-1}\rangle$ pode ser substituída pela porta de um qubit $\mathcal{CR}_k$ representada pela matriz

$$\mathcal{CR}_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i j_{k+l-1}}{2^k}} \end{pmatrix}$$

Quanto ao primeiro operador da figura 7.2 (porta de Hadamard), notamos que

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

Assim podemos escrever

$$|\psi\rangle = \mathcal{CR}_{n+1-l} \ldots \mathcal{CR}_2 \, H|j_l\rangle = \mathcal{CR}_{n+1-l} \ldots \mathcal{CR}_2 \, \mathcal{CR}_1 \, |+\rangle$$

onde

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

Definindo

$$\mathcal{PR}_{n+1-l} = \prod_{k=n+l-1}^{1} \mathcal{CR}_k$$

Temos

$$\mathcal{PR}_{n+1-l} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i j}{2^{n+1-l}}} \end{pmatrix}$$

Finalmente obtemos

$$|\psi\rangle = \mathcal{PR}_{n+1-l} \, |+\rangle = \frac{|0\rangle + e^{\frac{2\pi i j}{2^{n+1-l}}}|1\rangle}{\sqrt{2}}$$

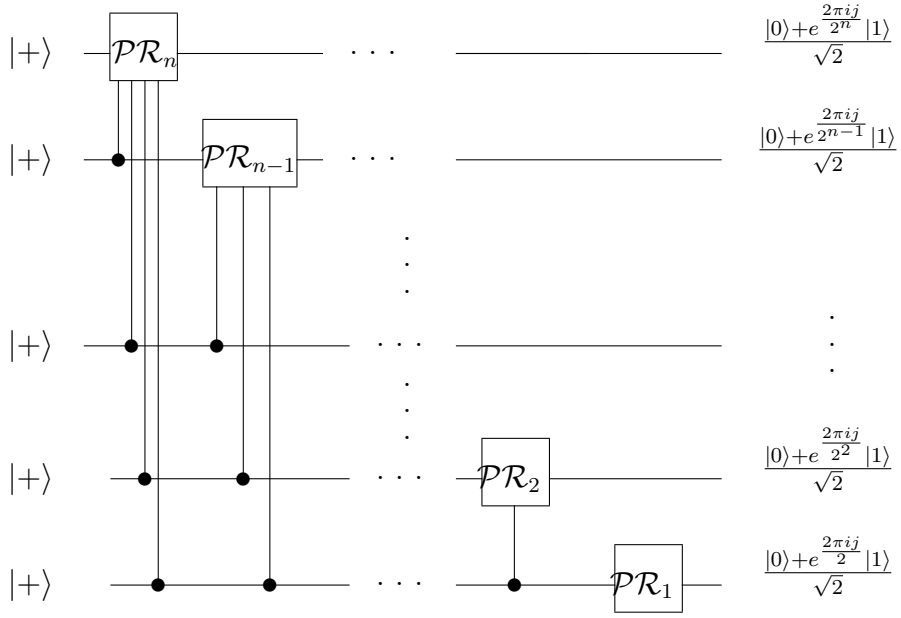

Figura 7.3: Circuito com portas $\mathcal{PR}$ e de Hadamard

Com os operadores $\mathcal{PR}_1$, $\mathcal{PR}_2$, ..., $\mathcal{PR}_n$ podemos construir o circuito da Figura 7.3, onde o processamento dos qubits do input ($|+\rangle$) se dá da esquerda para a direita e o output é igual (a menos da ordem) à expressão fatorada de $DFT|j\rangle$. Para ordenar a saída do circuito de cima para baixo na mesma ordem (da esquerda para a direita) da expressão fatorada de $DFT|j\rangle$ são necessárias $\frac{n}{2}$ trocas ("swaps"). Com isto a complexidade da DFT quântica passará a sua estimativa final, $O(n^3)$.

Com isto concluímos a representação da transformada de Fourier quântica em termos de circuito, isto é, decomposta "a nível de hardware" em termos de portas universais. Pudemos assim visualizar em detalhe um dos dois operadores centrais eminentemente quânticos do algoritmo de Shor, a saber, o operador DFT. Trata-se de um operador decisivo para o breakthrough propiciado pelo algoritmo.

Consideremos agora o outro operador eminentemente quântico da parte central, a porta U_f. Ainda que o considere como o "bottleneck" do algoritmo, Shor [39] desenvolve estimativas de complexidade para este operador, baseadas em circuitos conhecidos para exponenciação modular. Estes circuitos podem ser convertidos em um esquema de computação reversível resultando automaticamente em um circuito quântico.

Mais especificamente, Shor descreve um circuito quântico para um operador reversível com complexidade $O(n)$ e $O(n^3)$ em espaço de memória e tempo de execução, respectivamente, para o operador U_f dado por

$$U_f|x\rangle|0\rangle = |x\rangle|m^x \bmod N\rangle.$$

onde m, x, N são inteiros positivos com $n = \lceil \log_2 N \rceil$ e $\text{mdc}(m, N) = 1$. Este caso, em que m e N são primos entre si, é suficiente para estimar a complexidade associada ao operador U_f já que, em caso contrário, o algoritmo terminaria no passo 1, com a simples aplicação do algoritmo de Euclides, e este operador não se aplicaria.

Concluímos assim que a porta U_f não introduz complexidade super-polinomial no algoritmo, seja em espaço de memória ou em tempo de execução.

8
Considerações Finais

Ao longo desta dissertação tivemos a experiência de construir passo a passo nossa apreciação do algoritmo de fatoração de Shor como potencial breakthrough nas ciências da computação e como um exemplo emblemático de Computação Quântica, ou seja, como uma contribuição decisiva no esforço de síntese entre a Mecânica Quântica e as Ciências da Computação.

Nossos passos se deram em diferentes terrenos, cuja interatividade não nos era conhecida inicialmente, mas foi se nos apresentando cada vez mais nítida, em grande parte graças à orientação de que dispusemos: a Teoria dos Números, a Mecânica Quântica, as Ciências da Computação, a Teoria da Probabilidade e o próprio algoritmo em si, visto em diversas perspectivas: a do pseudo-código, a da representação por meio de circuitos, a da máquina de Turing, a da simulação clássica.

Nestas considerações finais nossa opção pelo termo "breakthrough" deve ser melhor qualificada, em especial com relação a considerações de eficiência. Com efeito, ao fornecer evidências de que computadores quânticos podem ser mais eficientes que computadores clássicos, Peter Shor trouxe à luz uma série de desafios e questionamentos. Como ocorre tipicamente em um "breakthrough", são novos desafios e questionamentos que constituem o cerne de uma nova contribuição para a ciência, ou mesmo para uma eventual ruptura de paradigma, cabendo aqui destacar:

- a possibilidade de rompimento de sistemas de criptografia consagrados em uso corrente em nível mundial (em especial o sistema RSA, baseado na dificuldade de fatoração de grandes inteiros)
- um estímulo para a pesquisa de algoritmos quânticos que permitam reduzir o tempo de execução de outros algoritmos clássicos (diríamos que Shor de certa forma "abriu um precedente")
- o questionamento da tese forte de Church-Turing, segundo a qual qualquer algoritmo pode ser simulado em uma máquina de Turing probabi-

lística com no máximo um aumento polinomial no número de operações elementares requeridas.

Enquanto o primeiro item acima tem uma importância eminentemente prática e o segundo se coloca entre o interesse prático e novas contribuições às ciências da computação, o terceiro é radicalmente mais impactante pois aponta para uma possível ruptura de paradigma, remetendo a nossa conjectura no capítulo 3 de que o insight original de Feynman [2] poderia levar à superação (em sentido dialético hegeliano) de conceitos da computação clássica fundamentados na máquina universal de Turing.

Em todos os desafios e questionamentos apontados, a ideia central subjacente é a de *eficiência*. Em particular, se a tese forte de Church-Turing estiver correta, então a execução eficiente de qualquer algoritmo em qualquer computador poderá ser simulada em uma máquina de Turing probabilística. Portanto para se verificar se um dado algoritmo pode ser executado eficientemente, bastará analisar sua exequibilidade (eficiente) em uma máquina de Turing probabilística.

O algoritmo de fatoração de Shor, em especial mediante a inserção de sua parte quântica, coloca isso em questão: a tese forte de Church-Turing e, com ela, a próprio reconhecimento da máquina de Turing probabilística como referência e critério universal de exequibilidade computacional eficiente.

Contudo, tal algoritmo em nada atinge a relevância central da máquina de Turing como modelo universal de computabilidade pura e simples. Em outras palavras, computadores quânticos e computadores clássicos permanecem compartilhando a máquina de Turing probabilística como padrão de referência para computabilidade. Um algoritmo é executável em um computador (clássico ou quântico) se e somente se o for em uma máquina de Turing. Entretanto, nem todos os algoritmos serão executáveis *com igual eficiência*, ou melhor, com mesma complexidade computacional.

Cabe aqui lembrar que muitas são as formas de se entender o conceito de eficiência computacional e nos restringiremos aqui a apenas três aspectos:

- tempo de processamento
- uso de espaço de memória
- uso de energia

Quanto ao item tempo de processamento (aspecto mais enfatizado ao longo desta dissertação), verificamos que o algoritmo de Shor contribui decisivamente para sua redução, de exponencial para polinomial, comparativamente aos algoritmos de computação clássica conhecidos, eminentemente por conta

dos operadores quânticos DFT e U_f. Quanto aos demais procedimentos do algoritmo, são todos clássicos e têm complexidade polinomial.

No capítulo 7, mediante a representação do algoritmo por meio de circuito, vimos como a transformada de Fourier discreta quântica (operador DFT) pode ser aplicada com complexidade polinomial, ao invés da complexidade exponencial requerida a nível clássico.

Já o operador U_f leva o vetor de estado $|\psi_1\rangle$ ao vetor de estado $|\psi_2\rangle$ mediante transformação física em nível quântico e medida direta — portanto mediante um procedimento físico e não por processamento numérico clássico. Não estudamos em detalhe este operador não trivial e optamos por considerá-lo como uma "caixa preta", ou melhor, uma "porta preta", que permanecerá fechada no âmbito desta dissertação. Contudo vale aqui lembrar que este operador não compromete a complexidade polinomial do algoritmo, conforme explicamos no seção 7.3 do capítulo 7.

Contudo vimos no capítulo 7 que o algoritmo de Shor é da classe BQP, dos problemas computacionais que podem ser resolvidos em tempo polinomial em um computador quântico com probabilidade de erro limitada. Tal classificação permanece bem respaldada, mesmo em face da "porta preta" U_f. Com efeito, U_f não introduz tempo de computação super polinomial no algoritmo [1].

Em resumo, tanto a parte não quântica quanto a parte quântica do algoritmo de fatoração de Shor têm complexidade polinomial, resultando na complexidade polinomial do algoritmo como um todo — em termos de tempo de execução.

No que se refere ao espaço em memória requerido, pode parecer que o algoritmo de Shor pague um alto preço por sua eficiência em termos de tempo de processamento.

Com efeito, vimos no capítulo 5 que em computação clássica L bits de memória são suficientes para armazenar um número inteiro a com $0 \leq a \leq 2^L - 1$, enquanto em Computação Quântica L *qubits* tensionados são necessários para tal. Entretanto o espaço de Hilbert que descreve L qubits tensionados, isto é, um tensor de L qubits, tem dimensão 2^L. Diante desse aumento exponencial na dimensão dos espaços envolvidos, podemos ainda considerar o algoritmo de Shor como um breakthrough em face dos algoritmos clássicos, apenas com base na redução (também exponencial) no tempo de execução?

A questão é em si pertinente ao tema desta dissertação, uma vez que fica faltando analisar o trade-off economia de tempo *versus* uso (eventualmente abusivo) de espaço na execução do algoritmo. A esse respeito vamos aqui nos limitar a apenas três considerações.

[1] Cf. Shor [39], Lavor [34], Lomonaco [19], Volovich [40].

Em primeiro lugar, a simples constatação de que igual número de bits clássicos e de qubits é suficiente para armazenar um dado número inteiro não bastaria para atribuirmos a ambos os procedimentos custos de mesma ordem de grandeza em nível de projeto, construção, manutenção e operação efetiva dos sistemas físicos envolvidos. A própria construção física do qubit é em si uma tarefa suficientemente desafiadora — e optamos por não abordá-la nesta dissertação.

Em segundo lugar, devemos considerar a classe de complexidade computacional PSPACE dos algoritmos que podem ser executados usando espaço polinomial sem limitação para o tempo de execução. Vimos que o algoritmo de Shor é da classe BQP e sabe-se que BQP $\subseteq$ PSPACE [2].

Em terceiro lugar, sabemos da Mecânica Quântica que em um meio físico de massa zero (o fóton) pode-se realizar o armazenamento de um número arbitrário de bits clássicos.

Assim constatamos que, comparativamente aos algoritmos clássicos conhecidos de fatoração, o algoritmo de Shor apresenta redução exponencial no tempo de execução com espaço adicional em termos de qubits ainda polinomial.

Finalmente, cabe considerar um aspecto menos encontradiço em avaliações de eficiência computacional: o uso de energia.

O consumo de energia em computação está ligado ao conceito de reversibilidade [3]. Em operadores computacionais irreversíveis parte da informação contida no input é perdida na operação, isto é, parte da informação é apagada. Por outro lado, em operadores reversíveis nenhuma informação é apagada, pois o input pode sempre ser recuperado a partir do output. Portanto, em computação reversível não há perda de informação.

Isto posto, a relação entre consumo de energia e irreversibilidade em computação é dada pelo princípio de Landauer, segundo o qual para apagar informação é necessário dissipar energia. Mais precisamente[4]:

Princípio de Landauer: Suponha que um computador apague um único bit de informação. A quantidade de energia dissipada no ambiente será pelo menos igual a $k_B\ T\ ln2$, onde k_B é a constante de Boltzmann e T é a temperatura ambiente.

Contudo, se todas as operações em um computador forem reversíveis, não haverá dispêndio de energia pois nenhum bit de informação é apagado em computação reversível. Supondo que as leis da física sejam fundamentalmente reversíveis, abre-se a possibilidade de explorar as leis da Mecânica Quântica

[2] Cf. Nielsen e Chuang [35] p. 41.
[3] Cf. Nielsen e Chuang [35], pp. 153-155.
[4] Cf. Nielsen e Chuang [35], p. 153.

(no espírito do insight seminal de Feynman [2]) para construir computadores quânticos com operadores reversíveis.

Todo esse cenário envolvendo avanços antes insuspeitados em termos de uso de tempo, espaço e energia em computação surge por conta da aplicação de procedimentos derivados da Mecânica Quântica, tanto na execução do cálculo como no armazenamento de dados em memória. Não por acaso, a entidade quântica *emaranhamento* se apresenta como peça chave no operador DFT quântico do algoritmo de Shor, precisamente no ponto onde, nesta dissertação, identificamos o centro do argumento a favor de sua complexidade polinomial. O emaranhamento é atualmente considerado como um candidato a reconhecimento como um elemento fundamental da Natureza, comparável a matéria, energia, informação e entropia[5].

Assim é aqui inevitável compartilhar com os primeiros físicos a sensação de $\theta\alpha\upsilon\mu\alpha$ quando, ao final desta dissertação, nos defrontamos com entidades como o fóton, com massa zero e memória espantosa, perante o operador físico-quântico instantâneo U_f, um autêntico "bottleneck" de todo o algoritmo que estudamos, perante uma nova entidade fundamental da natureza, o emaranhamento ...

Só nos resta encerrar esta dissertação voltando a sua epígrafe: "information is physical" ... indeed!

[5]Cf. Nielsen e Chuang [35], p. 12.

Referências bibliográficas

[1] C. F. Gauss, *Disquisitiones arithmeticae*, Asociación Costarricense de Historia y Filosofía de la Ciencia, 1995.

[2] R. P. Feynman, *Simulating physics with computers*, International journal of theoretical physics **21** (1982), no. 6, 467–488.

[3] Peter W Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM review **41** (1999), no. 2, 303–332.

[4] Paul Benioff, *The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines*, Journal of statistical physics **22** (1980), no. 5, 563–591.

[5] Paul Benioff, *Quantum mechanical hamiltonian models of turing machines*, Journal of Statistical Physics **29** (1982), no. 3, 515–546.

[6] Paul Benioff, *Quantum mechanical models of turing machines that dissipate no energy*, Physical Review Letters **48** (1982), no. 23, 1581.

[7] R. P. Feynman, *Quantum mechanical computers*, Foundations of physics **16** (1986), no. 6, 507–531.

[8] David Deutsch, *Quantum theory, the church–turing principle and the universal quantum computer*, Proc. R. Soc. Lond. A, vol. 400, The Royal Society, 1985, pp. 97–117.

[9] David Deutsch e Richard Jozsa, *Rapid solution of problems by quantum computation*, Proc. R. Soc. Lond. A, vol. 439, The Royal Society, 1992, pp. 553–558.

[10] André Berthiaume e Gilles Brassard, *The quantum challenge to structural complexity theory*, Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual, IEEE, 1992, pp. 132–137.

[11] André Berthiaume e Gilles Brassard, *Oracle quantum computing*, Journal of modern optics **41** (1994), no. 12, 2521–2535.

[12] E Bernstein e U Vazirani, *Proceedings of the 25th annual acm symposium on the theory of computing*, ACM, New York **11** (1993).

[13] Daniel R Simon, *On the power of quantum computation*, SIAM journal on computing **26** (1997), no. 5, 1474–1483.

[14] Carl Pomerance, *Fast, rigorous factorization and discrete logarithm algorithms*, Discrete algorithms and complexity, Elsevier, 1987, pp. 119–143.

[15] Daniel M Gordon, *Discrete logarithms in gf(p) using the number field sieve*, SIAM Journal on Discrete Mathematics **6** (1993), no. 1, 124–138.

[16] Arjen K Lenstra, Hendrik W Lenstra, Mark S Manasse, e John M Pollard, *The number field sieve*, The development of the number field sieve, Springer, 1993, pp. 11–42.

[17] Leonard M Adleman e Kevin S McCurley, *Open problems in number theoretic complexity, ii*, International Algorithmic Number Theory Symposium, Springer, 1994, pp. 291–322.

[18] Ronald L Rivest, Adi Shamir, e Leonard Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM **21** (1978), no. 2, 120–126.

[19] S. J. Lomonaco, *Shor's quantum factoring algorithm*, Proceedings of Symposia in Applied Mathematics, vol. 58, 2002, pp. 161–180.

[20] WG Teich, K Obermayer, e G Mahler, *Structural basis of multistationary quantum systems. ii. effective few-particle dynamics*, Physical Review B **37** (1988), no. 14, 8111.

[21] K Obermayer, WG Teich, e G Mahler, *Structural basis of multistationary quantum systems. i. effective single-particle dynamics*, Physical Review B **37** (1988), no. 14, 8096.

[22] Seth Lloyd, *A potentially realizable quantum computer*, Science **261** (1993), no. 5128, 1569–1571.

[23] Juan I Cirac e Peter Zoller, *Quantum computations with cold trapped ions*, Physical review letters **74** (1995), no. 20, 4091.

[24] David P DiVincenzo, *Two-bit gates are universal for quantum computation*, Physical Review A **51** (1995), no. 2, 1015.

[25] Tycho Sleator e Harald Weinfurter, *Realizable universal quantum logic gates*, Physical Review Letters **74** (1995), no. 20, 4087.

[26] Adriano Barenco, David Deutsch, Artur Ekert, e Richard Jozsa, *Conditional quantum dynamics and logic gates*, Physical Review Letters **74** (1995), no. 20, 4083.

[27] Isaac L Chuang, Y Yamamoto, e R Laflamme, *Decoherence and a simple quantum computer*, Tech. report, Los Alamos National Lab., NM (United States), 1995.

[28] Rolf Landauer, *Is quantum mechanics useful?*, Phil. Trans. R. Soc. Lond. A **353** (1995), no. 1703, 367–376.

[29] William G Unruh, *Maintaining coherence in quantum computers*, Physical Review A **51** (1995), no. 2, 992.

[30] G Massimo Palma, Kalle-Antti Suominen, e Artur K Ekert, *Quantum computers and dissipation*, Proc. R. Soc. Lond. A, vol. 452, The Royal Society, 1996, pp. 567–584.

[31] C. G. T. A. Moreira, F. E. B. Martínez, e N. C. Saldanha, *Tópicos de teoria dos números*, Sociedade Brasileira de Matemática, 2012.

[32] C. G. T. A. Moreira, F. E. B. Martínez, N. C. Saldanha, e E. Tengan, *Teoria dos números*, IMPA, 201.

[33] Godfrey Harold Hardy e Edward Maitland Wright, *An introduction to the theory of numbers*, Oxford university press, 1979.

[34] C. Lavor, L. R. U. Manssur, e R. Portugal, *Shor's algorithm for factoring large integers*, arXiv preprint quant-ph/0303175 (2003).

[35] M. A. Nielsen e I. L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, 2010.

[36] H. Häffner, C. F Roos, e R. Blatt, *Quantum computing with trapped ions*, Physics reports **469** (2008), no. 4, 155–203.

[37] M. H. Devoret, A. Wallraff, e J. M. Martinis, *Superconducting qubits: A short review*, arXiv preprint cond-mat/0411174 (2004).

[38] C. Nayak, S. H. Simon, A. Stern, M. Freedman, e S. D. Sarma, *Non-abelian anyons and topological quantum computation*, Reviews of Modern Physics **80** (2008), no. 3, 1083.

[39] P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on, Ieee, 1994, pp. 124–134.

[40] I. V. Volovich, *Quantum computing and shors factoring algorithm*, arXiv preprint quant-ph/0109004 (2001).

[41] R. Shankar, *Principles of quantum mechanics*, Plenum Press, 1994.

[42] D. J. Griffiths, *Introduction to quantum mechanics*, Prentice Hall, 2005.

[43] J. B. Conway, *Introduction to quantum mechanics*, Springer, 1978.

[44] G. Strang, *Linear algebra and its applications*, Harcourt Brace Jovanovich, 1988.

[45] Y. I. Manin, *Classical computing, quantum computing, and shor's factoring algorithm*, arXiv preprint quant-ph/9903008 (1999).

[46] S. Lloyd et al., *Universal quantum simulators*, Science-New York and Washington (1996), 1073–1077.

[47] A. Ekert e R. Jozsa, *Quantum computation and shor's factoring algorithm*, Reviews of Modern Physics **68** (1996), no. 3, 733–753.

[48] R. Landauer, *Information is physical*, Physics Today **44** (1991), no. 5, 23–29.