COMPUTAÇÃO QUÂNTICA: O ALGORITMO DE SHOR PARA FATORAÇÃO

MARCEL K. DE CARLI SILVA

Departamento de Ciência da Computação

Instituto de Matemática e Estatística

Universidade de São Paulo

Resumo. O problema da fatoração de inteiros em primos é um dos mais famosos em computação, tanto do ponto de vista teórico quanto prático. Apresentamos o modelo de computação quântico, introduzido nos anos 80, e o algoritmo de Shor, que fatora inteiros em primos eficientemente nesse modelo.

(Este trabalho está sendo desenvolvido junto com Carlos Henrique Cardonha, sob a orientação da Profa. Cristina Gomes Fernandes (IME-USP). O Carlos também fez uma submissão às Jornadas. A parte do trabalho submetida por ele complementa a nossa, abordando aspectos mais gerais do modelo quântico de computação, como as implicações dele na questão $\mathbf{P} = \mathbf{NP}$ e a relação das classes de complexidade clássicas e as quânticas.)

1. Introdução

Em 1900, em uma palestra marcante no Congresso Internacional de Matemáticos realizado em Paris, Hilbert postulou 23 problemas matemáticos, que tratam de temas diversos em matemática e áreas afins. O décimo problema na lista de Hilbert (*determination of the solvability of a diophantine equation*) pergunta se é possível determinar se uma equação diofantina arbitrária tem ou não solução por meio de um "processo finito":

> Given a diophantine equation with any number of unknown quantities and with rational numerical coefficients: to devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.

Esse problema pode ser postulado em uma linguagem mais atual como o seguinte: existe um algoritmo que, dada uma equação diofantina, determina se esta tem ou não solução?

Note que a questão postulada por Hilbert precede de décadas a invenção de computadores. Foi apenas nos anos 30 que tais questões foram formuladas e tratadas dentro do que ficou depois conhecido como *teoria da computabilidade*. Esta é a parte da teoria da computação especializada em lidar com esse tipo de questão.

Financiado parcialmente pela FAPESP 03/13237-7.

Foi nos anos 30, após um trabalho de Gödel em lógica, que a idéia de algoritmo começou a ser formalizada. Gödel [Göd31] introduziu o conceito de *função primitiva recursiva* como uma formalização dessa idéia. Church [Chu33, Chu36] introduziu o λ-*cálculo* e Kleene [Kle36] definiu o conceito de *funções recursivas parciais* e mostrou a equivalência entre esse e o λ-cálculo. Turing [Tur36, Tur37] por sua vez propôs a sua formalização da idéia de algoritmo: as chamadas *máquinas de Turing*. Nesses trabalhos, Turing mostrou também a equivalência do conceito de máquinas de Turing e de funções recursivas parciais de Church. Vale mencionar que o conceito de máquinas de Turing foi independentemente proposto por Post [Pos36], um professor de colegial de Nova Iorque. Cada uma dessas propostas diferentes do conceito de algoritmo é chamada de *modelo de computação*.

Foi Kleene [Kle52] quem chamou de *tese de Church* a afirmação de que todo modelo de computação *razoável* é equivalente ao da máquina de Turing. A afirmação é propositalmente vaga, pois visa capturar mesmo modelos que ainda venham a ser propostos, e cuja natureza não podemos prever. Por *razoável* entende-se um modelo que seja realista, no sentido de poder (mesmo que de maneira aproximada) ser construído na prática.

A teoria da computabilidade no fundo diferencia os problemas *decidíveis* (para os quais existe um algoritmo) dos *indecidíveis* (para os quais não existe um algoritmo). O surgimento dos computadores nas décadas de 30 e 40 aos poucos evidenciou uma diferença entre os problemas decidíveis: muitos parecem ser bem mais difíceis que outros, no sentido de que se conhece apenas algoritmos extremamente lentos para eles. Com isso, surgiu a necessidade de refinar a teoria de computabilidade para tentar explicar essas diferenças. Foi apenas nos anos 60 que a *teoria de complexidade*, que trata de tais questões, tomou corpo, com a formalização da idéia de "algoritmo eficiente", independentemente introduzida por Cobham [Cob65] e Edmonds [Edm65], e a proposta de reduções eficientes entre problemas [Kar72].

Foi nessa época que surgiram as definições das classes de complexidade **P** e **NP** e do conceito de **NP**-*completude*, que captura de certa maneira a dificuldade de se conseguir algoritmos eficientes para certos problemas. Grosseiramente, um problema em **NP** é dito **NP**-*completo* se qualquer outro problema da classe **NP** pode ser reduzido eficientemente a ele. A mais famosa questão na área de teoria da computação é se P é ou não igual a **NP**. Se for mostrado que algum problema **NP**-completo está em **P**, então tal questão é resolvida e fica provado que $\mathbf{P} = \mathbf{NP}$.

Um marco na teoria de complexidade é o *teorema de Cook* [Coo71, Lev73], que prova a existência de problemas **NP**-completos. Cook mostrou que o problema conhecido como SAT, de decidir se uma fórmula booleana em forma normal conjuntiva é ou não satisfatível, é **NP**-completo. Após o teorema de Cook e o trabalho de Karp [Kar72], que mostrou que vários outros problemas conhecidos de otimização combinatória eram **NP**-completos, essa teoria se desenvolveu amplamente, tendo estabelecido a dificuldade computacional de problemas das mais diversas áreas [GJ79].

Um problema muito famoso cuja complexidade continua em aberto, mesmo após várias décadas de esforço da comunidade no sentido de resolvê-lo, é o problema da *fatoração de inteiros*: dado um inteiro, determinar a sua fatoração em números primos. Recentemente, o seu parente próximo, o problema de decidir se um número inteiro é primo ou não, chamado de *problema da primalidade*, teve sua complexidade totalmente definida, com o algoritmo de Agrawal, Kayal e Saxena (AKS) [AKS02], que é aliás o assunto de um dos mini-cursos nesse colóquio. O algoritmo de AKS mostra que o problema da primalidade está na classe **P**, resolvendo com isso uma questão em aberto há anos. Não se sabe até hoje, no entanto, se há um algoritmo eficiente para resolver o problema da fatoração de inteiros!

Na verdade, a dificuldade computacional do problema da fatoração de inteiros tem sido usada de maneira crucial em alguns sistemas criptográficos bem-conhecidos. Se for descoberto um algoritmo eficiente para resolver o problema da fatoração, vários sistemas criptográficos importantes seriam quebrados, incluindo o famoso sistema RSA de chave pública.

O assunto de nossa iniciação científica — Computação Quântica — trata de um novo modelo de computação, o modelo quântico, que vem levantando questões intrigantes dentro da teoria de complexidade, e pode ter impactos práticos dramáticos no mínimo na área de criptologia. O modelo quântico de computação não infringe a validade da tese de Church, porém questiona a validade de uma versão mais moderna dessa, a chamada *tese de Church estendida*, que diz que todo modelo de computação razoável pode ser simulado *eficientemente* por uma máquina de Turing.

Pode-se dizer que a teoria de computação quântica iniciou-se nos anos 80, quando Feynman [Fey82] observou que um sistema quântico de partículas, ao contrário de um sistema clássico, parece não poder ser simulado eficientemente em um computador clássico e sugeriu um computador que explorasse efeitos da física quântica para contornar o problema. Desde então, até 1994, a teoria de computação quântica desenvolveu-se discretamente, com várias contribuições de Deutsch [Deu85, Deu89], Bernstein e Vazirani [BV97], entre outros, que colaboraram fundamentalmente para a formalização de um modelo computacional quântico.

Foi apenas em 1994 que a teoria recebeu um forte impulso e uma enorme divulgação. Isso deveu-se ao algoritmo de Shor [Sho94, Sho97], um algoritmo quântico eficiente para o problema da fatoração de inteiros, considerado o primeiro algoritmo quântico combinando relevância prática e eficiência. O algoritmo de Shor é uma evidência de que o modelo computacional quântico proposto pode superar de fato o modelo clássico, derivado das máquinas de Turing. O resultado de Shor impulsionou tanto a pesquisa prática, objetivando a construção de um computador segundo o modelo quântico, quanto a busca por algoritmos criptográficos alternativos e algoritmos quânticos eficientes para outros problemas difíceis. Essas e várias outras questões, relacionadas tanto com a viabilidade do modelo quântico quanto com as suas limitações, têm sido objeto de intensa pesquisa científica.

Do ponto de vista prático, busca-se descobrir se é ou não viável construir um computador segundo o modelo quântico que seja capaz de manipular números suficientemente grandes. Tal viabilidade esbarra em uma série de questões técnicas e barreiras físicas e tecnológicas. Já se tem notícia de computadores construídos segundo o modelo quântico, mas todos ainda de pequeno porte. Em 2001, por exemplo, foi construído um computador quântico com 7 *qubits* (o correspondente aos bits dos computadores tradicionais). Nesse computador, foi implementado o algoritmo de Shor que, nele, fatorou o número 15. Uma parte dos cientistas da computação acredita que a construção de computadores quânticos de maior porte será possível, enquanto outra parte não acredita nisso.

Do ponto de vista de teoria de complexidade, busca-se estabelecer a relação entre as classes de complexidade derivadas do modelo quântico e as classes de complexidade tradicionais. Também busca-se, claro, estabelecer a complexidade no modelo quântico de problemas bem-conhecidos, ou seja, busca-se por algoritmos quânticos eficientes para outros problemas relevantes.

Nesse trabalho, apresentamos o modelo quântico, um exemplo de algoritmo quântico simples e o algoritmo de Shor para fatoração. A parte do nosso trabalho de iniciação científica submetida por Carlos Henrique Cardonha complementa a nossa, abordando aspectos mais gerais do modelo quântico de computação, como as implicações dele na questão $\mathbf{P} = \mathbf{NP}$ e a relação das classes de complexidade clássicas e das quânticas.

Esperamos que estes dois trabalhos dêem uma visão do que é esta área nova e intrigante, e das suas potencialidades e dificuldades. O tema é multidisciplinar, no sentido de que depende de uma série de conceitos da mecânica quântica, e empresta a notação usada nessa área, o que dificulta um pouco a apresentação dos conceitos para pessoas de outras áreas, como computação e matemática. Um texto mais completo que estamos preparando dentro dessa iniciação científica, com esses resultados e outros, pode ser encontrado no endereço `http://www.ime.usp.br/~magal/quantum/`.

2. O modelo quântico de computação

2.1. **Bits quânticos.** Seja $\mathcal{H}_2$ um espaço de Hilbert de dimensão 2. Fixe uma base ortonormal $B_2 = \{|0\rangle, |1\rangle\}$ de $\mathcal{H}_2$. Um *qubit* ou *bit quântico* é um vetor unitário em $\mathcal{H}_2$, isto é, um vetor $|\phi\rangle \in \mathcal{H}_2$ é um qubit se

$$|\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle, \tag{1}$$

com $\alpha_0, \alpha_1 \in \mathbb{C}$ e $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Dizemos que os vetores $|0\rangle$ e $|1\rangle$ são os *estados básicos* e que o qubit $|\phi\rangle$ está numa *superposição*[1] de estados básicos. Chamamos ao coeficiente complexo α_j de *amplitude* do estado básico $|j\rangle$, para $j = 0, 1$.

No modelo clássico, se tivermos em mãos um bit b, podemos descobrir sem problemas se b vale 0 ou 1 e isso em nada afeta o valor de b, que permanece inalterado. Já no modelo quântico, se tentarmos descobrir o "valor" de um qubit $|\phi\rangle$, este é alterado irreversivelmente, sem contar que o resultado é probabilístico. Mais precisamente, ao medirmos o estado de um qubit $|\phi\rangle$ dado pela equação (1), enxergaremos o "valor" $|0\rangle$ com probabilidade $|\alpha_0|^2$ e o "valor" $|1\rangle$ com probabilidade $|\alpha_1|^2$. Se o "valor" observado for $|0\rangle$, o estado do qubit $|\phi\rangle$, imediatamente após a medição, será $|0\rangle$, e analogamente se o estado observado for $|1\rangle$. Note então que, apesar de um qubit armazenar uma quantidade não-enumerável de informação, só conseguimos obter dele dois "valores" através de medições.

Existem apenas duas portas lógicas operando sobre um bit clássico: a porta identidade e a negação. No modelo quântico de computação, qualquer transformação

[1] Contraste isto com o modelo clássico, onde um bit assume apenas um dos valores 0 ou 1.

unitária em $\mathcal{H}_2$ é uma porta quântica. Uma matriz $U \in \mathbb{C}^{2\times 2}$ é dita *unitária* se $U^*U = UU^* = I$, onde I é a matriz identidade e U^* é a transposta conjugada de U. Para trabalharmos com tais transformações, convencionamos que um qubit $|\phi\rangle$ dado pela equação (1) é representado pelo vetor coluna

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} = |\phi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle. \tag{2}$$

Assim, a matriz de Hadamard

$$H := \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3}$$

transforma o qubit $|\phi\rangle = |0\rangle$ no qubit

$$H|\phi\rangle = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big). \tag{4}$$

Será útil para nós convencionarmos uma representação gráfica para circuitos quânticos, indicando a ordem de aplicação de portas e medições. Por exemplo, o circuito ilustrado na figura 1 indica que a matriz de Hadamard H deve ser aplicada ao qubit $|\phi\rangle$ e depois o qubit resultante deve ser medido para obtermos o qubit $|\phi'\rangle$. Se $|\phi\rangle$ for inicializado com $|0\rangle$, então $|\phi'\rangle$ será $|0\rangle$ ou $|1\rangle$ equiprovavelmente, de acordo com a equação (4).

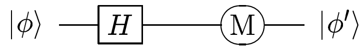

FIGURA 1. Um circuito quântico.

É interessante observar como as medições afetam o comportamento do circuito. Por exemplo, se inicializarmos $|\phi\rangle$ com $|0\rangle$, então $|\phi'\rangle$ no circuito quântico da figura 2 sempre será $|0\rangle$. Já no circuito da figura 3, o estado $|\phi'\rangle$ será $|0\rangle$ ou $|1\rangle$ equiprovavelmente.

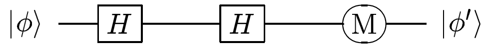

FIGURA 2. Mais um circuito.

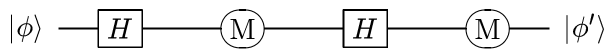

FIGURA 3. Circuito com medição intercalada.

2.2. **Registradores quânticos.** Seja $\mathcal{H}_{2^n}$ um espaço de Hilbert de dimensão 2^n. Denote por $\{0,1\}^n$ o conjunto das cadeias de caracteres de comprimento n sobre o alfabeto $\{0,1\}$ e fixe $B_{2^n} := \big\{|x\rangle : x \in \{0,1\}^n\big\}$ uma base ortonormal de $\mathcal{H}_{2^n}$. Por exemplo, para $n = 2$ temos $B_4 = \big\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\big\}$. Um registrador quântico de n qubits é um vetor unitário em $\mathcal{H}_{2^n}$, isto é, um vetor $|\phi\rangle \in \mathcal{H}_{2^n}$ é um registrador quântico de n qubits se

$$|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle, \tag{5}$$

com $\alpha_x \in \mathbb{C}$ para todo $x \in \{0,1\}^n$ e

$$\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1. \tag{6}$$

A nomenclatura para qubits se estende para os registradores: os estados $|x\rangle$ com $x \in \{0,1\}^n$ são os *estados básicos*, o qubit $|\phi\rangle$ é dito uma *superposição* de estados básicos e o coeficiente complexo α_x é chamado de *amplitude* do estado básico $|x\rangle$ para todo $x \in \{0,1\}^n$.

Será conveniente expressarmos o estado (5) como

$$|\phi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle, \tag{7}$$

onde estamos substituindo as cadeias de caracteres de $\{0,1\}^n$ pelos valores numéricos que essas cadeias representam, se interpretadas como representação binária de números. Por exemplo, para $n = 2$, temos

$$\begin{aligned} |\phi\rangle &= \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \\ &= \alpha|0\rangle + \beta|1\rangle + \gamma|2\rangle + \delta|3\rangle. \end{aligned}$$

Continuando a estender a notação de qubits para registradores, a representação por vetor-coluna do registrador quântico dado por (7) é

$$\begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{2^n-1} \end{pmatrix} = |\phi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle.$$

A seguinte questão surge naturalmente: dado um registrador $|\phi\rangle$ com n qubits cujos qubits, de menos para mais significativos, são $|\phi_0\rangle, |\phi_1\rangle, \ldots, |\phi_{n-1}\rangle$, qual é o estado quântico do registrador $|\phi\rangle$? A resposta é: $|\phi\rangle = |\phi_{n-1}\rangle \otimes |\phi_{n-2}\rangle \otimes \cdots \otimes |\phi_0\rangle$, onde $\otimes$ denota o produto tensorial, que definimos a seguir.

Sejam

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{pmatrix}$$

matrizes. O produto tensorial de A e B, denotado por $A \otimes B$, é definido como

$$A \otimes B = \begin{pmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{pmatrix}. \tag{8}$$

Assim, um registrador $|\phi\rangle$ cujos qubits são $|\phi_1\rangle$ e $|\phi_0\rangle$, com $|\phi_1\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ e $|\phi_0\rangle = \beta_0|0\rangle + \beta_1|1\rangle$, está no estado

$$|\phi\rangle = |\phi_1\rangle \otimes |\phi_0\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \otimes \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} = \begin{pmatrix} \alpha_0\beta_0 \\ \alpha_0\beta_1 \\ \alpha_1\beta_0 \\ \alpha_1\beta_1 \end{pmatrix}$$

e, portanto,

$$(9) \qquad \begin{aligned} |\phi\rangle &= \alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle + \alpha_1\beta_0|10\rangle + \alpha_1\beta_1|11\rangle \\ &= \alpha_0\beta_0|0\rangle + \alpha_0\beta_1|1\rangle + \alpha_1\beta_0|2\rangle + \alpha_1\beta_1|3\rangle. \end{aligned}$$

A medição de um registrador quântico funciona de modo semelhante à medição de qubits: se medirmos um registrador quântico $|\phi\rangle$ dado por (7), obtemos o estado básico $|x\rangle$ com probabilidade $|\alpha_x|^2$ e, imediatamente após a medição, o estado do registrador será $|x\rangle$, ou seja, toda a superposição que existia anteriormente foi irreversivelmente perdida.

Não é necessário, porém, medirmos todos os qubits do registrador: pode-se medir qubits individuais ou grupos de qubits. Por exemplo, se medirmos apenas o qubit $|\phi_1\rangle$ do registrador $|\phi\rangle$ descrito acima na equação (10), existem duas possibilidades de resposta:

- O valor observado é $|0\rangle$. Esse evento ocorre com probabilidade $|\alpha_0\beta_0|^2 + |\alpha_0\beta_1|^2 = |\alpha_0|^2$. Neste caso, o estado do registrador $|\phi\rangle$, imediatamente após a medição, será
$$|\phi'\rangle = \frac{\alpha_0\beta_0|00\rangle + \alpha_0\beta_1|01\rangle}{|\alpha_0|^2},$$
ou seja, projetou-se o estado $|\phi\rangle$ no subespaço gerado por $|00\rangle$ e $|01\rangle$, normalizando-se o resultado para obtermos um vetor unitário.
- O valor observado é $|1\rangle$. Esse evento ocorre com probabilidade $|\alpha_1\beta_0|^2 + |\alpha_1\beta_1|^2 = |\alpha_1|^2$. Neste caso, o estado do registrador $|\phi\rangle$, imediatamente após a medição, será
$$|\phi'\rangle = \frac{\alpha_1\beta_0|10\rangle + \alpha_1\beta_1|01\rangle}{|\alpha_1|^2},$$
ou seja, projetou-se o estado $|\phi\rangle$ no subespaço gerado por $|10\rangle$ e $|11\rangle$, normalizando-se o resultado para obtermos um vetor unitário.

Para o caso geral, suponha que seu registrador quântico com n qubits está no estado dado pela equação (5) e que você está medindo o qubit $|\phi_j\rangle$, onde o qubit menos significativo é $|\phi_0\rangle$ e o mais significativo é $|\phi_{n-1}\rangle$. Para cada cadeia de caracteres $x \in \{0,1\}^n$, escreva $x = x_{n-1} \cdots x_0$, com $x_k \in \{0,1\}$ para todo k. Existem duas possibilidades para o resultado da medição de $|\phi_j\rangle$:

- O valor observado é $|0\rangle$. A probabilidade de ocorrência desse evento é
$$p_0 = \sum \left\{ |\alpha_x|^2 : x \in \{0,1\}^n \text{ e } x_j = 0 \right\}.$$

Neste caso, o estado do registrador, imediatamente após a medição, será

$$|\phi'\rangle = \frac{\sum \{\alpha_x|x\rangle : x \in \{0,1\}^n \text{ e } x_j = 0\}}{p_0},$$

onde p_0 é simplesmente um fator de normalização.

- O valor observado é $|1\rangle$. A probabilidade de ocorrência desse evento é

$$p_1 = \sum \{|\alpha_x|^2 : x \in \{0,1\}^n \text{ e } x_j = 1\}.$$

Neste caso, o estado do registrador, imediatamente após a medição, será

$$|\phi'\rangle = \frac{\sum \{\alpha_x|x\rangle : x \in \{0,1\}^n \text{ e } x_j = 1\}}{p_1},$$

onde p_1 é simplesmente um fator de normalização.

O mecanismo de medição de um número arbitrário de qubits de um registrador é análogo.

2.3. **Reversibilidade.** Falemos agora de portas quânticas operando sobre mais de um qubit. Uma *porta quântica* sobre n qubits é uma função bijetora de $\mathcal{V}_{2^n}$ para $\mathcal{V}_{2^n}$, onde $\mathcal{V}_{2^n}$ é o conjunto de vetores unitários de $\mathcal{H}_{2^n}$. Em outras palavras, uma porta quântica é uma matriz unitária em $\mathcal{H}_{2^n}$.

Uma conseqüência disso é que toda porta quântica é reversível, isto é, existe uma bijeção entre o domínio e a imagem da função correspondente. Por exemplo, suponha que temos uma porta quântica U e um registrador $|\phi\rangle$ com dimensões compatíveis. Se aplicarmos U a $|\phi\rangle$ para obtermos $|\phi'\rangle = U|\phi\rangle$, então podemos obter o estado original $|\phi\rangle$ a partir de $|\phi'\rangle$ através da aplicação da porta quântica U^* (claramente U^* é unitária), pois $U^*|\phi'\rangle = U^*U|\phi\rangle = I|\phi\rangle = |\phi\rangle$. Em outras palavras, a aplicação de U não causa "perda de informação".

Esse não é o caso, por exemplo, da porta lógica $\vee$, o "ou" lógico do modelo clássico: suponha que temos dois bits clássicos a e b e que aplicamos a porta $\vee$ nesses bits, obtendo $c = a \vee b$. Se tivermos $c = 1$, não temos como obter os valores de a e b, pois podia valer que $a = 1$ e $b = 0$, ou que $a = 0$ e $b = 1$, ou ainda, que $a = 1$ e $b = 1$. Dizemos, por esse motivo, que a porta $\vee$ não é reversível.

Não é difícil, porém, construirmos uma "versão" da porta "ou" que seja reversível. Considere a função $f : \{0,1\}^3 \longrightarrow \{0,1\}^3$ dada por $f(x,y,z) = \big(x, y, z \oplus (x \vee y)\big)$, onde $\oplus$ denota a operação lógica "ou exclusivo". Em outras palavras, a aplicação da função f muda o valor do bit z se, e somente se, $x \vee y = 1$. Além disso, é trivial obtermos os valores originais de x, y e z, pois x e y fazem parte dos valores que saem da porta. É evidente que f é uma bijeção, de modo que f é reversível. A única diferença é que estamos armazenando informações a mais, o suficiente para sermos capazes de obter x e y a partir de $f(x,y,z)$.

Então existe uma matriz unitária U_f que "implementa" a função f. Dado um registrador $|\phi\rangle = |x\rangle \otimes |y\rangle \otimes |z\rangle$, onde $|x\rangle, |y\rangle$ e $|z\rangle$ são qubits, a porta U_f leva $|\phi\rangle$ ao estado $|\phi'\rangle = |x\rangle \otimes |y\rangle \otimes \big|z \oplus (x \vee y)\big\rangle$. Será mais conveniente usarmos a seguinte notação: dado um registrador $|\phi\rangle = |x,y,z\rangle$, a aplicação de U_f a $|\phi\rangle$ leva o estado deste registrador a $|\phi'\rangle = \big|x, y, z \oplus (x \vee y)\big\rangle$. Veja que estamos simplesmente separando os qubits individuais por vírgulas. Na verdade, podemos separar[2] grupos arbitrários de qubits num registrador, de acordo com a necessidade de clareza.

Esse "truque" de carregar informações a mais nas aplicações de portas lógicas pode ser utilizado para transformar qualquer porta lógica do modelo clássico numa porta reversível e que, portanto, pode ser implementada por uma matriz unitária no modelo quântico.

Na verdade, Bennett [Ben73] provou que qualquer circuito (no modelo clássico) pode ser convertido em reversível, ou seja, num circuito cujas portas são todas reversíveis, e que isso pode ser feito com um aumento no máximo polinomial no tamanho do circuito, isto é, no número de portas utilizadas. Isso implica, como veremos na seção seguinte, que todo algoritmo polinomial no modelo clássico pode ser implementado por um algoritmo polinomial no modelo quântico.

2.4. **Circuitos e algoritmos quânticos.** A notação para circuitos apresentada na seção sobre qubits se estende facilmente para circuitos operando sobre múltiplos qubits. Por exemplo, a figura 4 mostra um circuito operando sobre 3 qubits e com uma única porta, dada pela matriz U_f implementando a versão reversível da operação lógica "ou". De acordo com nossa especificação de U_f, temos $|x'\rangle = |x\rangle$, $|y'\rangle = |y\rangle$ e $|z'\rangle = \big|z \oplus (x \vee y)\big\rangle$.

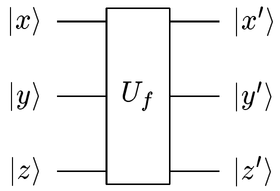

FIGURA 4. Circuito com porta "ou" reversível.

Seja $V_f \in \mathbb{C}^{2\times 2}$ uma matriz unitária. Então a notação utilizada no circuito da figura 5 indica que o operador V_f deve ser aplicado ao qubit $|y\rangle$ se, e somente se, $|x\rangle = |1\rangle$. Em outras palavras, o circuito da figura 5 é equivalente ao circuito da figura 6, onde

$$V'_f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & v_{11} & v_{12} \\ 0 & 0 & v_{21} & v_{22} \end{pmatrix} \qquad \text{e} \qquad V_f = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}.$$

A porta indicada no circuito da figura 5 é chamada *porta V_f controlada por $|x\rangle$*.

Para fazermos a análise do consumo de tempo de um algoritmo quântico, vamos nos basear em circuitos quânticos. Algoritmos quânticos devem ser expressos utilizando-se circuitos acíclicos. Estamos interessados em circuitos de tamanho polinomial no tamanho da entrada, onde o tamanho do circuito é simplesmente o número de portas quânticas utilizadas. Além disso, o circuito deve poder ser construído por

[2]Note que essa separação é apenas uma notação e não implica em nenhuma mudança efetiva.

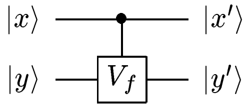

FIGURA 5. Circuito com porta V_f controlada por $|x\rangle$.

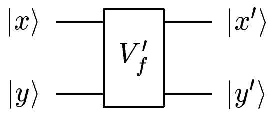

FIGURA 6. Circuito equivalente ao da figura 5.

um algoritmo polinomial. Em outras palavras, deve existir um algoritmo polinomial que, ao receber qualquer instância I de um dado problema, constrói o circuito que resolve o problema para a instância I. Essa exigência impede que o circuito construído contenha informações "difíceis de calcular" codificadas em sua estrutura: se o algoritmo de construção do circuito não precisasse ser limitado polinomialmente, então tal algoritmo poderia calcular, digamos em tempo exponencial, a solução da instância em questão e codificar isso no circuito, que poderia ter até tamanho logarítmico.

Ademais, cada uma das portas do circuito construído deve operar sobre, no máximo, um número previamente fixo de qubits. Esclarecendo melhor, fixe um inteiro k. Para todo n, o circuito construído para resolver uma instância de tamanho n deve utilizar portas quânticas que operam, no máximo, sobre k qubits. Essa exigência também é razoável, já que computações geralmente são realizadas localmente, isto é, a cada passo uma quantidade limitada de informações é processada.

Nesse contexto, um algoritmo quântico é dito polinomial se, para toda instância do problema, existe um circuito quântico construtível em tempo polinomial e utilizando portas operando sobre, no máximo, um número fixo de qubits, que resolve a instância do problema.

2.5. **O problema de Deutsch.** Vamos apresentar um algoritmo quântico para ver os conceitos apresentados em funcionamento.

Dizemos que uma função f é dada como uma *caixa preta* se só podemos obter informações acerca de f através de sua aplicação a elementos de seu domínio. O problema de Deutsch consiste no seguinte. Seja $f : \{0,1\} \longrightarrow \{0,1\}$ uma função dada como uma caixa preta. Determine se $f(0) = f(1)$ ou se $f(0) \neq f(1)$. Em outras palavras, determine se f é constante ou balanceada.

Para se resolver o problema com certeza no modelo clássico, são necessárias duas aplicações de f: é preciso usar a caixa preta de f duas vezes, para as entradas 0 e 1. Já no modelo quântico, este problema pode ser resolvido utilizando-se apenas uma chamada à caixa preta. Vamos mostrar um algoritmo quântico, devido a Cleve, Ekert, Macchiavello e Mosca [CEMM98], que resolve o problema no modelo quântico com uma única chamada à caixa preta.

Seja U_f a transformação unitária de dimensão 4 que leva $|x, y\rangle$ a $|x, y \oplus f(x)\rangle$. No modelo quântico, U_f é a nossa caixa preta.

O circuito que resolve o problema está descrito na figura 7. Vamos detalhar melhor o funcionamento do algoritmo.

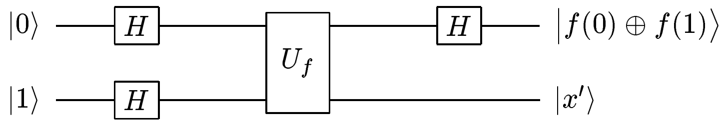

FIGURA 7. Circuito para o problema de Deutsch.

Começamos com um registrador $|\phi_0\rangle$ de 2 qubits inicializado com $|0,1\rangle$.

Primeiro aplicamos a transformação de Hadamard aos 2 qubits do registrador, obtendo

$$\begin{aligned} |\phi_1\rangle &= \big(H|0\rangle\big) \otimes \big(H|1\rangle\big) = \frac{1}{2}\left[\Big(|0\rangle + |1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big)\right] \\ &= \frac{1}{2}\left[|0\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right] + \frac{1}{2}\left[|1\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right]. \end{aligned}$$

Neste ponto a caixa preta U_f é aplicada a $|\phi_1\rangle$ para obtermos $|\phi_2\rangle$:

$$\begin{aligned} |\phi_2\rangle &= U_f|\phi_1\rangle \\ &= \frac{1}{2}\left\{U_f\left[|0\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right]\right\} + \frac{1}{2}\left\{U_f\left[|1\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right]\right\}. \end{aligned} \tag{10}$$

Vamos escrever $|\phi_2\rangle$ de outra maneira. Queremos mostrar que

$$U_f\left[|x\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right] = (-1)^{f(x)}\left[|x\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right] \tag{11}$$

para $x \in \{0,1\}$. Temos

$$\begin{aligned} |\psi\rangle &= U_f\left[|x\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right] = U_f\Big[|x,0\rangle - |x,1\rangle\Big] \\ &= \big|x, f(x)\big\rangle - \big|x, 1 \oplus f(x)\big\rangle \end{aligned}$$

- Se $f(x) = 0$, então

$$|\psi\rangle = |x,0\rangle - |x,1\rangle = |x\rangle \otimes \Big(|0\rangle - |1\rangle\Big) = (-1)^{f(0)}\left[|x\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right].$$

- Se $f(x) = 1$, então

$$|\psi\rangle = |x,1\rangle - |x,0\rangle = |x\rangle \otimes \Big(|1\rangle - |0\rangle\Big) = (-1)^{f(1)}\left[|x\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\right].$$

Provamos então a equação (11), de modo que,

$$\begin{aligned}
|\phi_2\rangle &= \frac{1}{2}\left\{(-1)^{f(0)}\Big[|0\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\Big]\right\} + \frac{1}{2}\left\{(-1)^{f(1)}\Big[|1\rangle \otimes \Big(|0\rangle - |1\rangle\Big)\Big]\right\} \\
&= \frac{1}{2}\Big[\Big((-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big)\Big] \\
&= \frac{1}{2}\Big[\Big((-1)^{f(0)}|0\rangle + (-1)^{f(1)}(-1)^{f(0)}(-1)^{f(0)}|1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big)\Big] \\
&= \frac{(-1)^{f(0)}}{2}\Big[\Big(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle\Big) \otimes \Big(|0\rangle - |1\rangle\Big)\Big] \\
&= (-1)^{f(0)}\left\{\ \Big[\frac{1}{\sqrt{2}}\Big(|0\rangle + (-1)^{f(0)\oplus f(1)}|1\rangle\Big)\Big] \otimes \Big[\frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big)\Big]\ \right\}.
\end{aligned}$$

Agora podemos aplicar a transformação de Hadamard ao primeiro qubit de $|\phi_2\rangle$ para obter

$$|\phi_3\rangle = (-1)^{f(0)}\left\{\Big[\big|f(0) \oplus f(1)\big\rangle\Big] \otimes \Big[\frac{1}{\sqrt{2}}\Big(|0\rangle - |1\rangle\Big)\Big]\ \right\}.$$

Uma medição do primeiro qubit de $|\phi_3\rangle$ fornece agora o valor de $f(0) \oplus f(1)$ e portanto o algoritmo descobre com certeza se f é constante ou balanceada através de uma única aplicação da caixa preta.

3. O algoritmo de fatoração de Shor

O algoritmo de Shor resolve o problema da fatoração de inteiros em primos e consome tempo polinomial no tamanho da entrada. Apresentamos a seguir os detalhes fundamentais do funcionamento deste algoritmo.

3.1. **Visão geral do algoritmo.** O algoritmo de Shor [Sho97] é um algoritmo quântico[3] que, dado um n inteiro composto ímpar que não é potência de primo, devolve um fator[4] de n com probabilidade limitada de erro.

As restrições para o valor de n não representam problema algum. De fato, é trivial encontrar um fator de um número par. Além disso, não é difícil pensar num algoritmo clássico eficiente que decide se $n = a^k$, para inteiros a e $k > 1$, e que devolve a e k neste caso.

Ademais, podemos verificar em tempo polinomial se n é composto, utilizando o algoritmo AKS. Outra opção é executar testes probabilísticos de primalidade um

[3]Na verdade, Shor [Sho97] sugere que algumas partes do seu algoritmo, como o cálculo do máximo divisor comum, sejam executadas num computador clássico, a fim de economizar tempo. É evidente que tais partes, após a devida conversão que as torne reversíveis, podem ser realizadas eficientemente por um computador quântico. Mas não há motivos para não se usar computadores clássicos para procedimentos que não façam uso de propriedades exclusivas do modelo quântico.

[4]Um fator de n é um divisor não-trivial de n.

número suficiente de vezes. Na prática, isso é mais eficiente, pois os testes probabilísticos são, em geral, mais simples e rápidos que o AKS. O teste de Miller-Rabin [Mil75, Mil76, Rab80, CLRS01] é uma ótima escolha para esta verificação, por ser de fácil implementação e ter complexidade de tempo $O(\lg^3 n)$, com uma constante pequena escondida pela notação assintótica.

Como n é produto de no máximo $\lg n$ inteiros, o algoritmo de Shor pode ser utilizado para resolver o problema da fatoração em tempo polinomial no tamanho da entrada.

O algoritmo de Shor baseia-se numa redução do problema da busca de um fator de n ao problema da busca do período de uma seqüência. Como a redução utiliza aleatorização, é possível que ela falhe, isto é, que nenhum fator de n seja encontrado. Porém, a probabilidade de ocorrência deste evento é limitada. Na seção 3.2 apresentamos essa redução e limitamos a probabilidade de falha.

Na seção 3.3, mostramos um algoritmo quântico eficiente para a busca do período da seqüência gerada pela redução. Esse algoritmo utiliza a transformada quântica de Fourier, que pode ser implementada eficientemente, como mostramos na seção 3.4.

O algoritmo de busca de período apresentado na seção 3.3 pode ser facilmente generalizado para buscar eficientemente o período de qualquer seqüência. Mais formalmente, dado um oráculo U_f que computa uma função f de $\{0, \ldots, 2^m - 1\}$ em $\{0, \ldots, 2^a - 1\}$ com período r, a generalização do algoritmo faz uma única chamada a U_f e usa um circuito quântico de tamanho polinomial em m para descobrir r com probabilidade limitada de erro.

3.2. **Redução à busca do período.** Seja n um inteiro composto ímpar que não é potência de primo. Vamos mostrar como reduzir o problema de encontrar um fator de n ao problema de encontrar o período de uma função. Essa redução utiliza aleatorização, de modo que precisaremos limitar a probabilidade de falha do procedimento.

Algoritmo ENCONTRA-FATOR (n)

```
1    escolha um inteiro 1 < x < n aleatoriamente
2    se mdc(x, n) > 1
3        então devolva mdc(x, n)
4    seja r o período da função f(a) = x^a mod n
5    se r for ímpar ou x^{r/2} ≡ −1 (mod n)
6        então o procedimento falhou
7    devolva mdc(x^{r/2} + 1, n)
```

Primeiro note que, se o algoritmo executa a linha 3, então o valor devolvido de fato é um fator de n.

Já se $\text{mdc}(x, n) = 1$, então x está em $\mathbb{Z}_n^*$, o grupo multiplicativo módulo n, de modo que a função $f(a) = x^a \bmod n$ é periódica com período dado pela ordem de x, módulo n. Isto é, o período r é o tamanho do subgrupo de $\mathbb{Z}_n^*$ gerado por x. Equivalentemente, r é o menor inteiro positivo tal que $x^r \equiv 1 \pmod{n}$.

Suponha que r é par, de modo que $(x^{r/2}+1)(x^{r/2}-1) \equiv 0 \pmod{n}$, ou seja, n divide o produto $(x^{r/2}+1)(x^{r/2}-1)$. Se tivermos $x^{r/2} \not\equiv -1 \pmod{n}$, então n não divide $x^{r/2}+1$, o primeiro fator do produto. Como r é o menor inteiro positivo tal que $x^r \equiv 1 \pmod{n}$, temos também $x^{r/2} \not\equiv 1 \pmod{n}$, de modo que n não divide $x^{r/2}-1$. Mas então os fatores de n devem estar separados entre $x^{r/2}+1$ e $x^{r/2}-1$. Logo, $\mathrm{mdc}(x^{r/2}+1, n)$ é um fator de n, como queremos.

Resta limitarmos a probabilidade de falha do procedimento, ou seja, dado um inteiro $1 < x < n$ escolhido aleatoriamente com probabilidade uniforme, precisamos limitar a probabilidade de que r, a ordem de x, módulo n, seja ímpar ou satisfaça $x^{r/2} \equiv -1 \pmod{n}$ se for par.

Suponha que a fatoração de n em primos é dada por $n = \prod_{i=1}^{m} p_i^{k_i}$, onde p_i é primo para todo i e $m > 1$, já que n não é potência de primo. Para cada i, seja $n_i := {p_i}^{k_i}$, de modo que $n = n_1 \cdots n_m$. Sejam $1 < x < n$ um inteiro,

$$r \text{ a ordem de } x, \text{ módulo } n,$$

e

$$r_i \text{ a ordem de } x, \text{ módulo } n_i,$$

para $i = 1, \ldots, m$. Pelo teorema chinês do resto, a equação $x^r \equiv 1 \pmod{n}$ é equivalente ao sistema

$$\begin{aligned} x^r &\equiv 1 \pmod{n_1} \\ x^r &\equiv 1 \pmod{n_2} \\ &\vdots \\ x^r &\equiv 1 \pmod{n_m}. \end{aligned} \tag{12}$$

Sabemos que r_i é o menor inteiro positivo tal que $x^{r_i} \equiv 1 \pmod{n_i}$. Ademais, temos $x^r \equiv 1 \pmod{n_i}$ se, e somente se, r é múltiplo de r_i. Como r é o menor inteiro positivo tal que $x^r \equiv 1 \pmod{n}$, segue que

$$r = \mathrm{mmc}(r_1, \ldots, r_m). \tag{13}$$

Seja

$$r_i = 2^{c_i} q_i, \text{ com } q_i \text{ ímpar}, \tag{14}$$

para $i = 1, \ldots, m$. É fácil ver que r é ímpar se, e somente se, r_i é ímpar para todo i, isto é, se, e somente se, $c_i = 0$ para todo i.

Suponha agora que r_i é par para algum i. Então r é par. Vamos descobrir em que condições temos $x^{r/2} \equiv -1 \pmod{n}$. Novamente, pelo teorema chinês do resto, a equação $x^{r/2} \equiv -1 \pmod{n}$ é equivalente ao sistema

$$\begin{aligned} x^{r/2} &\equiv -1 \pmod{n_1} \\ x^{r/2} &\equiv -1 \pmod{n_2} \\ &\vdots \\ x^{r/2} &\equiv -1 \pmod{n_m}. \end{aligned} \tag{15}$$

Suponha que existam $i, j \in \{1, \ldots, m\}$ tais que $c_i > c_j$. Então $r = 2r_j u$ para algum inteiro u, pela equação (13). Segue que $x^{r/2} \equiv x^{r_j u} \pmod{n_j}$. Mas então temos

$x^{r/2} \equiv 1 \pmod{n_j}$, de modo que $x^{r/2} \not\equiv -1 \pmod{n}$. Portanto, para que r seja par e $x^{r/2} \equiv -1 \pmod{n}$, é necessário que $c_1 = c_2 = \cdots = c_m > 0$.

Estabelecemos assim que, se o procedimento falha, então $c_1 = \cdots = c_m$. Vamos limitar a probabilidade de ocorrência desse evento, dada a escolha aleatória de x.

Pelo teorema chinês do resto, existe uma bijeção entre $\mathbb{Z}_n$ e o produto cartesiano $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}$:

$$\mathbb{Z}_n \ni x \leftrightarrow (x_1, \ldots, x_m) \in \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_m}. \tag{16}$$

Assim, escolher um inteiro $0 \leq x < n$ aleatoriamente é equivalente a escolher, independentemente para cada $1 \leq i \leq m$, um inteiro $0 \leq x_i < n_i$. Vamos supor que $\mathrm{mdc}(x, n) = 1$, já que a redução não se aplica se $\mathrm{mdc}(x, n) > 1$. Então $x \in \mathbb{Z}_n^*$. É fácil ver que $x \in \mathbb{Z}_n^*$ se, e somente se, $x_i \in \mathbb{Z}_{n_i}^*$ para todo i. Portanto estamos escolhendo aleatoriamente um $x_i \in \mathbb{Z}_{n_i}^*$, independentemente, para cada $i = 1, \ldots, m$:

$$\mathbb{Z}_n^* \ni x \leftrightarrow (x_1, \ldots, x_m) \in \mathbb{Z}_{n_1}^* \times \cdots \times \mathbb{Z}_{n_m}^*, \tag{17}$$

Para cada $i = 1, \ldots, m$, o grupo $\mathbb{Z}_{n_i}^*$ é cíclico, pois $n_i = {p_i}^{k_i}$ com p_i primo. Seja g_i um gerador de $\mathbb{Z}_{n_i}^*$, para cada i. Seja

$$x_i = g_i^{l_i}, \text{ com } 0 \leq l_i < \phi(n_i), \text{ para } i = 1, \ldots, m, \tag{18}$$

onde $\phi(n_i) := \left|\mathbb{Z}_{n_i}^*\right|$ é a função totiente de Euler. Estamos então escolhendo aleatoriamente um $0 \leq l_i < \phi(n_i)$ para cada i, independente e uniformemente:

$$\mathbb{Z}_n^* \ni x \leftrightarrow (g_1^{l_1}, \ldots, g_m^{l_m}) \in \mathbb{Z}_{n_1}^* \times \cdots \times \mathbb{Z}_{n_m}^*, \tag{19}$$

Para $i = 1, \ldots, m$, seja

$$\phi(n_i) = 2^{d_i} s_i, \text{ com } s_i \text{ ímpar},$$

e lembre-se que

$$r_i = 2^{c_i} q_i \text{ é a ordem de } x_i, \text{ módulo } n_i, \text{ com } q_i \text{ ímpar.}$$

Como $\phi(n_i) = \phi\big(p_i^{k_i}\big) = p_i^{k_i - 1}(p_i - 1)$ e p_i é ímpar, então $d_i > 0$.

Pelo teorema de Lagrange, a ordem do subgrupo de $\mathbb{Z}_{n_i}^*$ gerado por x divide a ordem do grupo $\mathbb{Z}_{n_i}^*$. Em outras palavras, r_i divide $\phi(n_i)$, e portanto $c_i \leq d_i$. Sabemos que r_i é o menor inteiro positivo tal que $g_i^{l_i r_i} \equiv 1 \pmod{n_i}$ e que $l_i r_i = 2^{c_i} q_i l_i$ é múltiplo de $\phi(n_i) = 2^{d_i} s_i$. Se l_i é ímpar, então devemos ter $c_i = d_i$, pois $q_i l_i$ é ímpar. Já se l_i é par, então necessariamente teremos $c_i < d_i$: se $c_i = d_i$, então $r_i/2$ também é inteiro e $l_i r_i/2$ também é múltiplo de $\phi(n_i)$, um absurdo. Portanto, a probabilidade de que $c_i = c$, para qualquer c, é limitada por $1/2$, já que $0 \leq l_i < \phi(n_i)$ e $\phi(n_i)$ é par.

Concluímos então que a probabilidade de falha dessa redução, ou, na verdade, a probabilidade de que $c_1 = \cdots = c_m$ é, no máximo, $1 - 1/2^{m-1} \leq 1/2$, pois $m > 1$ e a escolha de cada c_i é independente de todas as outras.

3.3. **Busca do período.** Seja n um inteiro composto ímpar que não é potência de primo e seja $1 < x < n$ um inteiro relativamente primo a n. Vamos apresentar um algoritmo quântico que descobre, com probabilidade limitada de erro, a ordem de x, módulo n, que é o período da seqüência

$$\langle x^0 \bmod n, x^1 \bmod n, x^2 \bmod n, \ldots \rangle. \tag{20}$$

Seja $\beta := \lfloor \lg n \rfloor + 1$ o número de bits da representação binária de n e seja $q = 2^l$ a única potência de 2 tal que $n^2 \leq q < 2n^2$. Vamos precisar de um registrador $|\phi\rangle$ com $l + \beta$ qubits. Os l primeiros qubits formam o primeiro sub-registrador. Os β qubits restantes farão parte do segundo sub-registrador. Todos os qubits do registrador $|\phi\rangle$ devem ser inicializados com $|0\rangle$.

Após a aplicação da transformação de Hadamard a cada um dos qubits do primeiro sub-registrador, o estado do registrador $|\phi\rangle$ será

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, 0\rangle. \tag{21}$$

Seja U_f a transformação unitária que, para todo $0 \leq a < q$, leva um estado básico $|a, 0\rangle$, ao estado $|a, x^a \bmod n\rangle$. É fácil pensar num algoritmo clássico para a exponenciação modular que consome $O(\beta^3)$ operações sobre bits. Então, para todo n, existe um circuito com $O(\beta^3)$ portas, cada uma operando sobre no máximo um número fixo de qubits, que efetua a transformação U_f. Em outras palavras, U_f pode ser implementada eficientemente[5].

Aplicando a transformação U_f ao registrador $|\phi\rangle$, obtemos o estado

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a, x^a \bmod n\rangle. \tag{22}$$

Medindo o estado[6] do segundo sub-registrador de $|\phi\rangle$, obtemos algum $x^b \bmod n$, onde $0 \leq b < q$. Com isso, o estado do registrador $|\phi\rangle$ colapsa para uma superposição dos estados básicos de (22) cujos β qubits menos significativos representam $x^b \bmod n$.

Seja r a ordem de x, módulo n. Então os valores de $0 \leq a < q$ tais que $x^a \equiv x^b \pmod{n}$ são da forma $a_0 + jr$, com $0 \leq a_0 < r$, já que a seqüência (20) é periódica com período r. A medição do segundo sub-registrador em (22) selecionará os seguintes valores de a, no primeiro sub-registrador: $a_0, a_0 + r, a_0 + 2r, \ldots, a_0 + (A-1)r$, onde

[5]Não vamos nos deter em detalhes de como tal circuito deve ser implementado. A transformação ingênua do algoritmo de exponenciação modular em reversível gera um circuito que utiliza $O(\beta^3)$ qubits, pois é necessário armazenar qubits adicionais para manter a reversibilidade. Além disso, esta operação é o gargalo do algoritmo de Shor, ou seja, é a operação assintoticamente mais cara. Por esse motivo, Shor [Sho97] mostra uma implementação mais eficiente em termos de espaço, sem com isso piorar o consumo de tempo.

[6]Esta medição é desnecessária. A execução do algoritmo funciona da mesma forma, independente desta medição. Porém, decidimos incluí-la aqui por uma questão de clareza.

$A = \lceil q/r \rceil$. O estado do registrador $|\phi\rangle$ será então

$$\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |a_0 + jr, x^b \bmod n\rangle.$$

De agora em diante, vamos ignorar o segundo sub-registrador. Então o estado de $|\phi\rangle$ será

$$\frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} |a_0 + jr\rangle. \tag{23}$$

Note que os estados básicos de $|\phi\rangle$ que podem ser obtidos numa medição estão uniformemente espaçados a partir de a_0, com espaço r.

Seja M_q a transformação unitária[7] dada por

$$M_q : |x\rangle \longrightarrow \frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \exp(2\pi i x y/q)|y\rangle. \tag{24}$$

Vamos mostrar, na seção 3.4, que esta transformação pode ser implementada eficientemente por um circuito quântico. A aplicação de M_q ao registrador $|\phi\rangle$, dado por (23), gera o estado

$$\begin{aligned} |\phi\rangle &= \frac{1}{\sqrt{A}} \sum_{j=0}^{A-1} \left[\frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \exp\left\{2\pi i (a_0 + jr) y/q\right\} |y\rangle \right] \\ &= \frac{1}{\sqrt{qA}} \sum_{y=0}^{q-1} \left[\exp\{2\pi i a_0 y/q\} \sum_{j=0}^{A-1} \exp\{2\pi i j r y/q\} |y\rangle \right]. \end{aligned}$$

Então a amplitude de um estado básico $|y\rangle$ é

$$\frac{1}{\sqrt{qA}} \exp\{2\pi i a_0 y/q\} \sum_{j=0}^{A-1} \exp\{2\pi i j r y/q\}, \tag{25}$$

de modo que a probabilidade de obtenção de $|y\rangle$ numa medição de $|\phi\rangle$ é

$$p_y = \frac{1}{qA} \left| \sum_{j=0}^{A-1} \exp\{2\pi i j r y/q\} \right|^2. \tag{26}$$

Aqui vamos apenas analisar o caso em que r é uma potência de 2, de modo que $A = q/r$. Os demais casos seguem a mesma idéia porém são mais técnicos.

[7]É fácil ver que M_q é uma matriz de Vandermonde.

Suponha que y não é múltiplo de A. Então ry/q não é inteiro, de modo que $\exp\{2\pi i ry/q\} \neq 1$. Pela fórmula da soma de uma progressão geométrica, temos

$$\begin{aligned}\sum_{j=0}^{A-1} \exp\{2\pi i jry/q\} &= \sum_{j=0}^{A-1} \big(\exp\{2\pi i ry/q\}\big)^j \\ &= \frac{\big(\exp\{2\pi i ry/q\}\big)^A - 1}{\exp\{2\pi i ry/q\} - 1} \\ &= \frac{\exp\{2\pi i y\} - 1}{\exp\{2\pi i ry/q\} - 1} = 0,\end{aligned}$$

pois $A = q/r$.

Suponha agora que y é um múltiplo de A. Então ry/q é inteiro, de modo que $\exp\{2\pi i jry/q\} = 1$ para todo $0 \leq j < A$. Então

$$\sum_{j=0}^{A-1} \exp\{2\pi i jry/q\} = A.$$

Concluímos que a probabilidade de obtenção de $|y\rangle$ na medição do registrador $|\phi\rangle$ no estado (25) é

$$p_y = \begin{cases} 1/r, & \text{se } y \text{ é múltiplo de } q/r \\ 0, & \text{caso contrário.} \end{cases} \tag{27}$$

Assim, uma medição de $|\phi\rangle$ nos fornece um $y = cq/r$, com $0 \leq c < r$ escolhido equiprovavelmente. Teremos então um valor de y satisfazendo $y/q = c/r$, onde c e q são conhecidos. Se $\mathrm{mdc}(c, r) = 1$, então basta obter a fração irredutível correspondente a y/q para chegarmos ao período r. A probabilidade de obtenção de um $0 \leq c < r$ com $\mathrm{mdc}(c, r) = 1$ é $\phi(r)/r$. Pode-se provar [HW54] que existe uma constante δ tal que $\phi(r)/r > \delta/\log\log r$. Assim, a probabilidade de falha deste procedimento é, no máximo, $1 - \delta/\log\log r$.

Se repetirmos o procedimento acima $z := \log\log r/\delta$ vezes, a probabilidade de falha passará a ser $(1 - 1/z)^z \leq 1/e$, de modo que a probabilidade de sucesso é, no mínimo, $1 - 1/e$, uma constante. Portanto, obtemos o período r com probabilidade limitada inferiormente por uma constante.

3.4. **A transformada quântica de Fourier.** Vamos ver agora que a transformação unitária M_q, dada por (24), pode ser implementada eficientemente sempre que q for uma potência de 2. Ou seja, para todo $q = 2^m$, vamos mostrar que existe um circuito de tamanho polinomial em m, utilizando apenas portas quânticas que operam sobre um número fixo de qubits, cuja aplicação a um registrador com m qubits é equivalente à aplicação da matriz M_q.

Primeiro vamos escrever o estado

$$\frac{1}{\sqrt{q}} \sum_{y=0}^{q-1} \exp(2\pi i xy/q)|y\rangle \tag{28}$$

de uma forma mais conveniente. Considere $q = 2^m$, com m um inteiro positivo. Denotaremos por $(x_{m-1} \cdots x_0)_2$ a representação binária de $0 \leq x < 2^m$, ou seja, $x = \sum_{j=0}^{m-1} x_j 2^j$, com $x_j \in \{0,1\}$ para todo j. Além disso, utilize $(0.\, x_1 \cdots x_p)_2$ para denotar a representação binária de $0 \leq x < 1$, isto é, $x = \sum_{j=1}^{p} x_j 2^{-j}$, com $x_j \in \{0,1\}$ para todo j.

Então o estado (28) não é emaranhado e pode ser fatorado como

$$\begin{aligned} &\left[\frac{1}{\sqrt{2}}\Big(|0\rangle + \exp\left\{2\pi i(0.\, x_0)_2\right\}|1\rangle\Big)\right] \otimes \\ &\quad\left[\frac{1}{\sqrt{2}}\Big(|0\rangle + \exp\left\{2\pi i(0.\, x_1 x_0)_2\right\}|1\rangle\Big)\right] \otimes \cdots \otimes \\ &\qquad\left[\frac{1}{\sqrt{2}}\Big(|0\rangle + \exp\left\{2\pi i(0.\, x_{m-1} \cdots x_0)_2\right\}|1\rangle\Big)\right]. \end{aligned} \tag{29}$$

Para mostrar que o estado quântico (28) é o mesmo que o estado (29), vamos mostrar que, para todo estado básico $|y_{m-1} \cdots y_0\rangle$, temos

$$\begin{aligned} &\exp\{2\pi i xy/2^m\}|y_{m-1} \cdots y_0\rangle = \\ &\quad\Big(\exp\left\{2\pi i(0.\, x_0)_2 y_{m-1}\right\}|y_{m-1}\rangle\Big)\otimes \\ &\quad\Big(\exp\left\{2\pi i(0.\, x_1 x_0)_2 y_{m-2}\right\}|y_{m-2}\rangle\Big) \otimes \cdots \otimes \\ &\qquad\Big(\exp\left\{2\pi i(0.\, x_{m-1} \cdots x_0)_2 y_0\right\}|y_0\rangle\Big), \end{aligned} \tag{30}$$

onde o lado direito da equação (30) mostra como se forma o estado básico $|y_{m-1} \cdots y_0\rangle$ em (29). Será então suficiente mostrar que

$$\begin{aligned} &\exp\{2\pi i xy/2^m\} \\ &\quad= \exp\left\{2\pi i(0.\, x_0)_2 y_{m-1}\right\}\ \exp\left\{2\pi i(0.\, x_1 x_0)_2 y_{m-2}\right\} \cdots \\ &\qquad\quad \exp\left\{2\pi i(0.\, x_{m-1} \cdots x_0)_2 y_0\right\} \\ &\quad= \exp\Big\{2\pi i\big[(0.\, x_0)_2 y_{m-1} + (0.\, x_1 x_0)_2 y_{m-2} + \cdots + \\ &\qquad\quad (0.\, x_{m-1} \cdots x_0)_2 y_0\big]\Big\}. \end{aligned} \tag{31}$$

Observe que

$$\frac{yx}{2^m} = \frac{1}{2^m} \sum_{k=0}^{m-1} \left[y_k 2^k \sum_{k=0}^{m-1} x_k 2^k \right] = \sum_{j=0}^{m-1} \left[y_j \sum_{k=0}^{m-1} x_k 2^{k-m+j} \right]. \tag{32}$$

Na equação (31), o termo $xy/2^m$ aparece multiplicando $2\pi i$. Então apenas a parte fracionária de $xy/2^m$ é relevante: se $xy/2^m = u + r$, com $0 \leq r < 1$ e $u \in \mathbb{Z}$, então $\exp(2\pi i xy/2^m) = \exp(2\pi i r)$. Na equação (32), para $k \geq m - j$, o valor 2^{k-m+j} é inteiro, de modo que podemos reescrever (32) como

$$\frac{yx}{2^m} = \sum_{j=0}^{m-1} \left[y_j u_j + y_j \sum_{k=0}^{m-j-1} x_k 2^{k-m+j} \right], \tag{33}$$

onde u_j é um inteiro. É fácil verificar agora que

$$\sum_{k=0}^{m-j-1} x_k 2^{k-m+j} = (0.\,x_{m-j-1}\cdots x_0)_2.$$

Mas então

$$xy/2^m = u + y_{m-1}(0.\,x_0)_2 + y_{m-2}(0.\,x_1x_0)_2 + \cdots + y_0(0.\,x_{m-1}\cdots x_0)_2, \tag{34}$$

onde u é um inteiro. A equação (31) segue imediatamente da equação (34), de modo que fica provado que o estado (28) pode ser fatorado como (29).

Considere o circuito quântico apresentado na figura 8, referente ao caso em que $m = 4$.

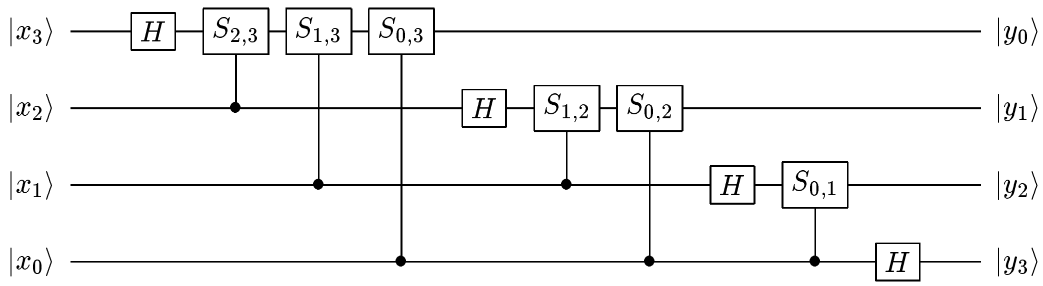

FIGURA 8. Circuito para a transformada quântica de Fourier, com $m = 4$.

Nesta figura, a matriz $S_{j,k}$, para $j < k$ é definida por

$$S_{j,k} = \begin{bmatrix} 1 & 0 \\ 0 & \exp\{2\pi i/2^{k-j+1}\} \end{bmatrix}. \tag{35}$$

É muito fácil ver como este circuito se estende para qualquer m. Para cada qubit $|x_k\rangle$, aplicamos a matriz de Hadamard, seguida de k portas $S_{j,k}$, onde a porta $S_{j,k}$ é controlada por $|x_j\rangle$ para todo $0 \leq j < k$.

Para ver que esse circuito de fato calcula a transformada quântica de Fourier, vamos analisar sua operação sobre o qubit $|x_3\rangle$ na figura 8. Após a aplicação da matriz de Hadamard, o estado deste qubit será $|0\rangle + \exp\left\{2\pi i(0.\,x_3)_2\right\}|1\rangle$. Note que estamos desprezando o fator de normalização $1/\sqrt{2}$, para maior clareza. Depois da aplicação da porta $S_{2,3}$, o estado passa a ser $|0\rangle + \exp\left\{2\pi i(0.\,x_3x_2)_2\right\}|1\rangle$. Aplicando agora a porta $S_{1,3}$, teremos $|0\rangle + \exp\left\{2\pi i(0.\,x_3x_2x_1)_2\right\}|1\rangle$ e, por fim, após $S_{0,3}$ o estado será $|0\rangle + \exp\left\{2\pi i(0.\,x_3\cdots x_0)_2\right\}|1\rangle$. Mas isso é justamente $|y_0\rangle$, conforme a equação (29). Repetindo esses cálculos com os outros qubits, obteremos os estados desejados, de acordo com a equação (29).

Note que os qubits da saída deste circuito estão na ordem inversa. É óbvio que isso não representa qualquer problema para nós.

Observe também que o circuito utiliza $m(m+1)/2$ portas, o que é quadrático em m. Assim, a transformada quântica de Fourier pode ser implementada por um circuito de tamanho limitado por $\Theta(m^2)$.

Referências

[AKS02] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Preprint. Disponível em `http://www.cse.iitk.ac.in/news/primality.pdf`, 2002.

[Ben73] C.H. Bennett. Logical reversibility of computation. *IBM J. Res. Develop.*, 17:525–532, 1973.

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[CEMM98] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.*, 454(1969):339–354, 1998.

[Chu33] A. Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 25:839–864, 1933.

[Chu36] A. Church. An unsolvable problem of elementary number theory. *Annals of Mathematics*, 58:345–363, 1936.

[CLRS01] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, Cambridge, MA, second edition, 2001.

[Cob65] A. Cobham. The intrinsic computational difficult of functions. In Y. Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science*, pages 24–30. North-Holland, 1965.

[Coo71] S. Cook. The complexity of theorem proving procedures. In *Proc. 3rd ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[Deu85] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. Roy. Soc. London Ser. A*, 400(1818):97–117, 1985.

[Deu89] D. Deutsch. Quantum computational networks. *Proc. Roy. Soc. London Ser. A*, 425(1868):73–90, 1989.

[Edm65] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[EJ96] A. Ekert and R. Jozsa. Quantum computation and Shor's factoring algorithm. *Rev. Mod. Phys.*, 68(3), 1996.

[Fey82] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6 & 7):467–488, 1982.

[Göd31] K. Gödel. On formally undecidable propositions of Principia Mathematica and related systems. *Monatshefte fur Math. und Physik*, 38:173–198, 1931.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* Freeman, 1979.

[HW54] G.H. Hardy and E.M. Wright. *An introduction to the theory of numbers.* Oxford, at the Clarendon Press, 1954. 3rd ed.

[Kar72] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972)*, pages 85–103. Plenum, New York, 1972.

[Kle36] S. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112:727–742, 1936.

[Kle52] S. Kleene. *Introduction to Metamathematics.* D. Van Nostrand, Princeton, NJ, 1952.

[Lev73] L.A. Levin. Universal sorting problems. *Problems of Information Transmission*, 9:265–266, 1973.

[Mil75] G.L. Miller. Riemann's hypothesis and tests for primality. In *Seventh Annual ACM Symposium on Theory of Computing (Albuquerque, N.M., 1975)*, pages 234–239. Assoc. Comput. Mach., New York, 1975.

[Mil76] G.L. Miller. Riemann's hypothesis and tests for primality. *J. Comput. System Sci.*, 13(3):300–317, 1976. Working papers presented at the ACM-SIGACT Symposium on the Theory of Computing (Albuquerque, N.M., 1975).

[MM04] D.C. Marinescu and G.M. Marinescu. Lectures on quantum computing. `http://www.cs.ucf.edu/~dcm/Fall2003Class-QC/QCTextBookIndex.htm`, 2004.

[MR95] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge University Press, Cambridge, 1995.

[Pos36] E. Post. Finite combinatory process. *Journal of Symbolic Logic*, 1:103–105, 1936.

[Pre04] J. Preskill. Lecture notes for physics 219/computer science 219. `http://www.theory.caltech.edu/people/preskill/ph229`, 2004.

[Rab80] M.O. Rabin. Probabilistic algorithm for testing primality. *J. Number Theory*, 12(1):128–138, 1980.

[RP00] E. Rieffel and W. Polak. Introduction to quantum computing. *ACM Computing Surveys*, 32(3):300–335, 2000.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science (Santa Fe, NM, 1994)*, pages 124–134. IEEE Comput. Soc. Press, Los Alamitos, CA, 1994.

[Sho97] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Tur36] A. Turing. On computable numbers with an application to the entscheidungsproblem. *Proc. London Math. Soc.*, 2:230–265, 1936.

[Tur37] A. Turing. Rectifications to 'on computable numbers...'. In *Proc. London Mathematical Society*, volume 4, pages 544–546, 1937.

Instituto de Matemática e Estatística, Universidade de São Paulo, Rua do Matão 1010, 05508–900 São Paulo, SP

Endereços Eletrônicos: `magal@ime.usp.br`

URL: `http://www.ime.usp.br/~magal/quantum`