Introdução à Computação Quântica

Amanda Castro Oliveira, Renato Portugal

LNCC/MCT

28 a 31/01/2008

Sumário

Introdução: O que é Computação Quântica

- Estudo de dispositivos computacionais baseados na Mecânica Quântica desperta grande interesse.
- Nasce a Computação Quântica!
- É um domínio recente que combina três áreas bem conhecidas: Matemática, Física e Computação.
- É o estudo das tarefas que podem ser realizadas pelo processamento da informação contida em sistemas quânticos [1].
- Motivação: Sistemas Quânticos podem processar informação de forma mais eficaz.
- É um modelo de computação distinto da computação clássica. Se baseia nas leis da Mecânica Quântica diferentemente da computaçõ clássica que tem seu funcionamento baseado nas leis da Física Clássica. No lugar dos bits entram os q-bits além de novas portas lógicas que realizam diversas tarefas.

Introdução: O que é Computação Quântica

- 3 grandes resultados:
  - Código de correção de erros [2, 3, 4];
  - Criptografia Quântica [5, 6];
  - Algoritmo de Shor[7];
- O desafio estava lançado! Surgem duas grandes frentes de trabalho: Hardware e Software.

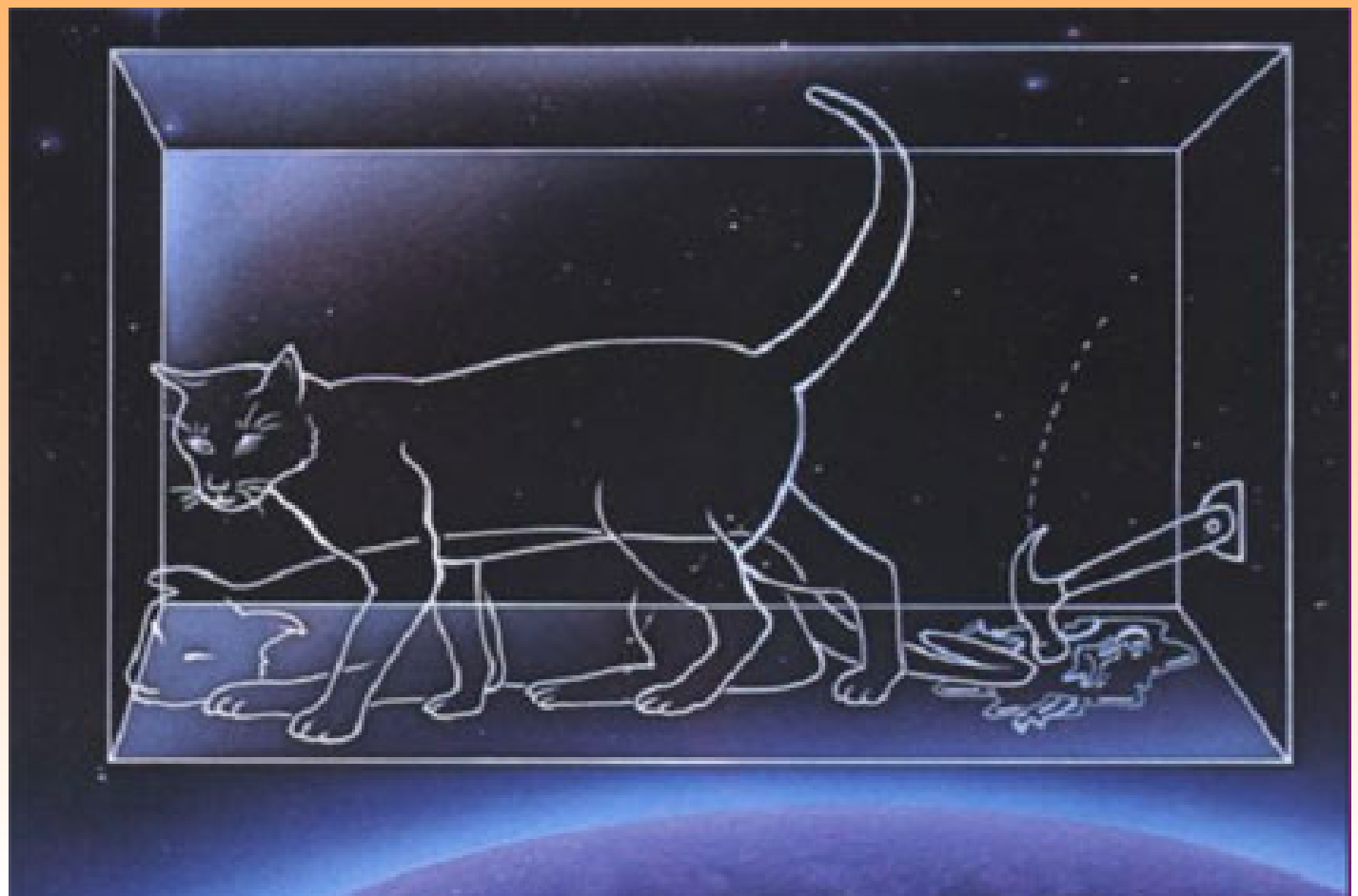
Introdução: O que é Computação Quântica

- Experimentos bem controlados com no máximo 10 q-bits, por enquanto!!
- Até o momento não há resultado algum que inviabilize o computador quântico.

- Algoritmo de Shor é o primeiro! Fatora números inteiros grandes eficientemente.
- O paralelismo dos sistemas quânticos vem de graça, mas extrair a resposta certa nem sempre é simples!
- Precisamos investigar os casos clássicos bem sucedidos e verificar se podem ser mais eficientes no caso quântico.

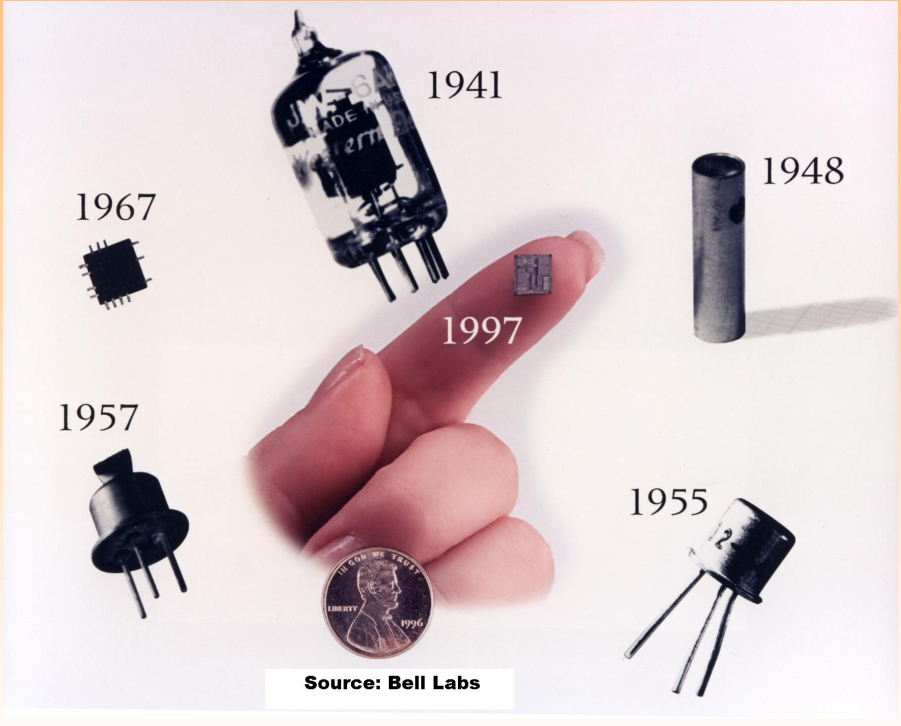
Histórico da Computação Quântica

- Virada do século XX até a década de 20 surge a Mecânica Quântica. Estudo do comportamento de sistemas microscópicos. Regras são simples, mas não intuitivas. Ex. Possibilidade de um gato estar vivo e morto simultâneamente!!



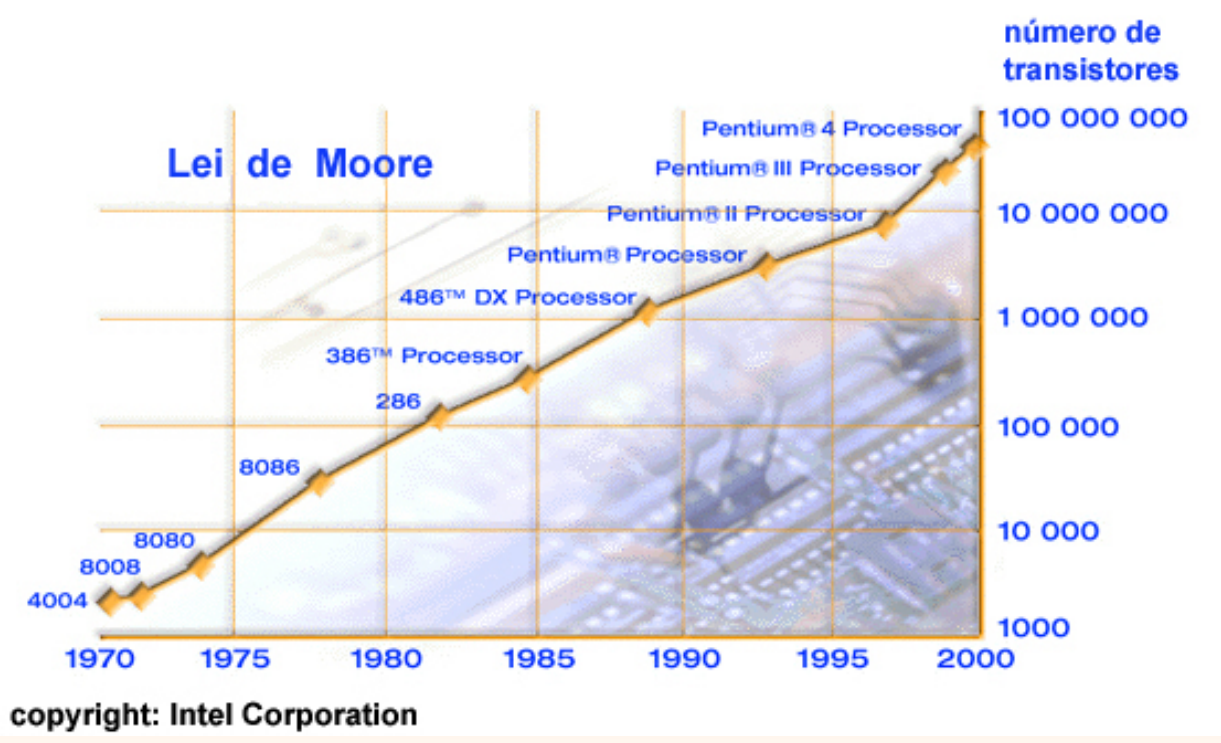
Histórico da Computação Quântica

- Em 1936 surge o grande paradigma da ciência da computação a Máquina de Turing. Logo depois surgem os primeiros computadores construídos a partir de componentes eletrônicos .
- Em 1947 John Bardeen, Walter Brattain e Will Shockley desenvolvem o transistor. Grande progresso!!



Histórico da Computação Quântica

- Em 1965 Gordon Moore estabelece uma lei que diz que o número de transistores usados em um circuito integrado dobra a cada dois anos. Isto é, daqui a dois anos vamos comprar um chip com o dobro da capacidade de processamento pelo mesmo preço que pagamos hoje!!



Histórico da Computação Quântica

- Com a constante miniaturização dos chips cogita-se que tecnologia convencional utilizada para a sua fabricação esbarraria nas dificuldades impostas pela redução do tamanha dos componentes.
- Os efeitos quânticos começam a interferir no funcionamento dos componentes à medida que eles diminuem.
- Uma possível solução: Um novo paradigma da computação. Usar a mecânica quântica para fazer computação.

Histórico da Computação Quântica

- A partir da década de 1970 várias técnicas que permitem o controle individual de sistemas quânticos foram desenvolvidas. Início do hardware quântico!!
- Em 1980 Paul Benioff foi o precursor a aplicar a teoria quântica a computadores.
- Mas foi em 1982 que Richard Feynman considerou que efeitos quânticos poderiam oferecer algo realmente novo. Ele mostrou como um sistema quântico poderia ser usado para fazer cálculos. Além de explicar como tal máquina seria capaz de agir como um simulador para a física quântica. Um físico poderia realizar experiências de física quântica usando um computador baseado na mecânica quântica.

Histórico da Computação Quântica

- Já em 1985 David Deutsch descreve o primeiro modelo teórico para a Máquina de Turing Quântica mostrando que qualquer processo físico, em princípio, poderia ser perfeitamente modelado por um computador quântico. Assim, um computador quântico teria habilidades naturais muito além daquelas de qualquer computador clássico tradicional. Ele apresentou um algoritmo que utiliza apenas operações quânticas capaz de resolver um certo problema matemático de modo mais eficiente que qualquer algoritmo clássico possível.
- Esse algoritmo ficou esquecido até 1989, quando Deutsch introduziu o modelo de circuitos quânticos. Com uma linguagem similar à linguagem de circuitos lógicos/ digitais amplamente utilizada começaram a aparecer outros algoritmos quânticos.

Histórico da Computação Quântica

- Em 1993 Charles Bennett e colaboradores da IBM mostram que a teleportação é de fato possível, desde que se destrua a amostra original!!

(top, left) Richard Jozsa, William K. Wootters, Charles H. Bennett. (bottom, left) Gilles Brassard, Claude Crépeau, Asher Peres. Photo: André Berthiaume.

Histórico da Computação Quântica

- Em 1994 Peter Shor apontou um método, usando computadores quânticos, para separar em várias partes um importante problema matemático na teoria dos números, chamado fatorização. Ele mostrou de que modo um conjunto de operações matemáticas, projetadas especificamente para um computador quântico, poderia ser organizado para permitir que tal máquina **fatorasse números enormes de um modo extremamente rápido, muito mais rápido do que é possível nos computadores convencionais**!!!

Histórico da Computação Quântica

- Em 1996 Lov K. Grover cria um algoritmo quântico que realiza buscas em um banco de dados desordenado que é quadraticamente mais rápido que os análogos clássicos.
- 1998 Isaac Chuang coordena um grupo em Berkeley que desenvolve o primeiro computador quântico de 1 q-bit!
- 2001 Um grupo da IBM desenvolve um computador quântico de 7 q-bits e fatoram o número 15 usando o algoritmo de Shor!!

O Computador Clássico

Um computador clássico é uma máquina construída, baseada nas leis da Física Newtoniana, que, em linhas gerais, lê um certo conjunto de dados, codificado em 0 e 1, executa cálculos e gera uma saída também codificada em 0 e 1.

- 0 e 1 são os bits: estados que podem ser representados fisicamente através do potencial elétrico.
  - 0: baixo potencial, por ex. 0.13 volts;
  - 1: alto potencial, por ex. 2.74 volts.
- Um computador é um dispositivo que calcula uma função $f : \{0, ..., N-1\} \rightarrow \{0, ..., N-1\}$, onde $N = 2^n$ (n é o número de bits da memória do computador).
- A cada conjunto de n bits de entrada, corresponde a um único conjunto de n bits de saída, caracterizando f como uma função.

O Computador Clássico

- f pode ser decomposta em blocos elementares que são implementados fisicamente por **transistores** e outros **componentes eletrônicos**.
- Esses blocos elementares são as portas lógicas AND, OR e NOT - portas universais.
- Basta a porta NOT e só uma das outras AND ou OR para que seja possível fazer qualquer operação!!

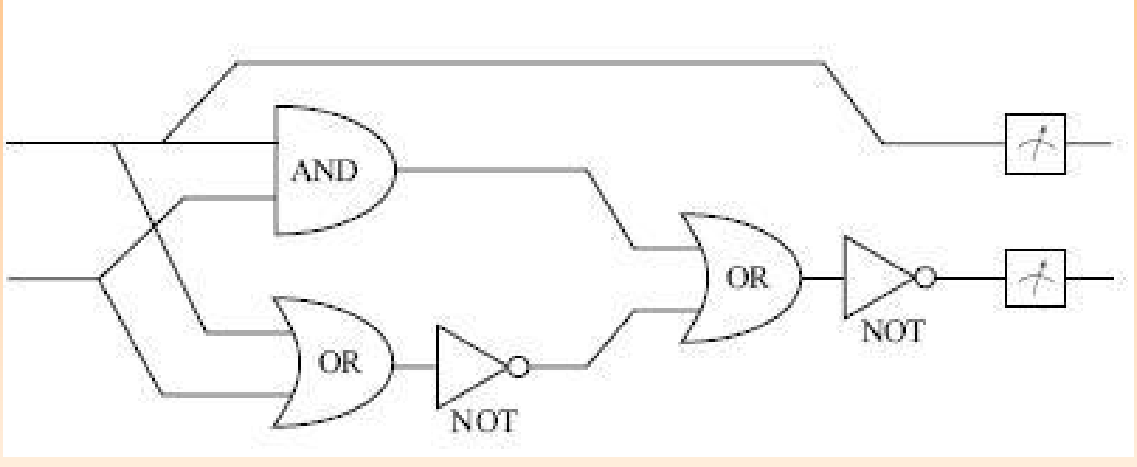


Figura: Circuito para realizar a soma mod2 de dois números de 1 bit

O Computador Clássico

$\oplus$	0	1
0	0	1
1	1	0

Tabela: Soma mod2 de 2 bits

Este é um circuito **IRREVERSÍVEL**, pois as portas AND e OR são irreversíveis. Desta forma dada a saída não temos como recuperar os dados de entrada.

AND	0	1
0	0	0
1	0	1

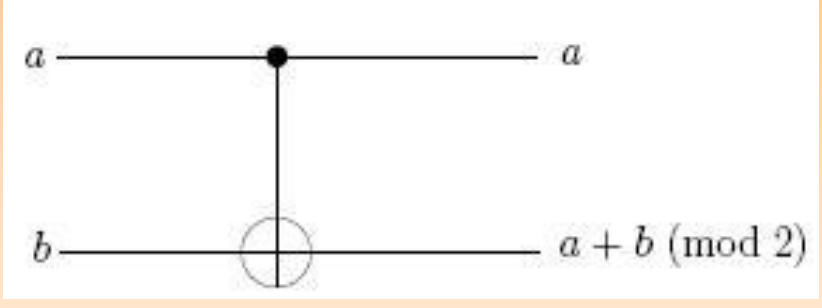
Tabela: A porta AND é irreversível.

OR	0	1
0	0	1
1	1	1

Tabela: A porta OR é irreversível.

O Computador Clássico

- Para transformarmos esse circuito em um circuito equivalente **reversível** vamos usar a porta CNOT.
- Esta porta CNOT é uma porta NOT controlada pelo valor do bit superior que é chamado bit de controle.
- O valor do bit inferior de saída nos fornece $a \oplus b$.



b \ a	0	1
0	0	1
1	1	0

Figura: Porta CNOT

O Computador Clássico

Podemos generalizar a porta CNOT usando 2 bits de controle. Neste caso temos a chamada porta Toffoli queťpode ser usada para a obtenção da porta AND reversível.

000	$\longrightarrow$	000
001	$\longrightarrow$	001
010	$\longrightarrow$	010
011	$\longrightarrow$	011
100	$\longrightarrow$	100
101	$\longrightarrow$	101
110	$\longrightarrow$	111
111	$\longrightarrow$	110

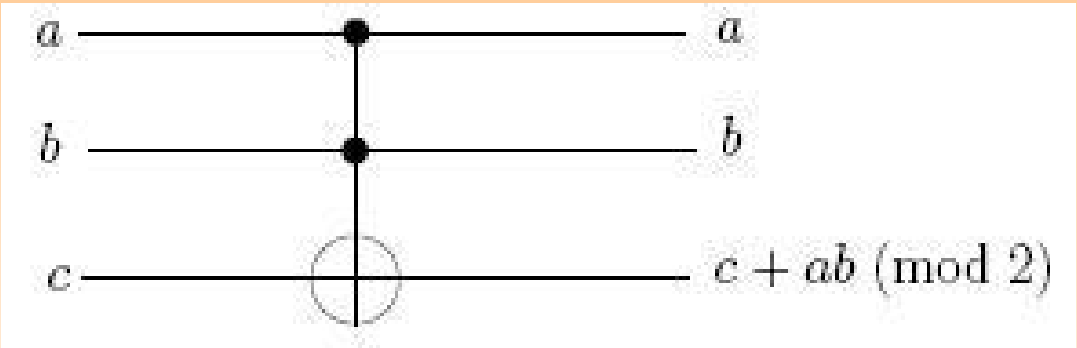


Figura: Porta Toffoli

Teremos a porta AND reversível fazendo $c = 0$ na porta Toffoli.

O Computador Clássico - Últimas considerações

- A todo o momento fazemos bifurcações nos fios duplicando a informação!! Há várias maneiras de se fazer isso!! Quanticamente isto é impossível!!
- Se o computador tem n bits de entrada, temos 2^n entradas possíveis, então temos 2^n saídas possíveis. Logo podemos obter $(2^n)^{2^n}$ funções. Que podem ser obtidas usando as portas universais.
- A velocidade com que um computador calcula essas funções depende da quantidade de portas usadas no circuito. Se o número de portas cresce **polinomialmente** com n o circuito é dito **eficiente**. Mas se esse número cresce **exponencialmente** dizemos que o circuito é **ineficiente**.
- Todos os cálculos clássicos poderão ser feitos em um computador quântico desde que as portas clássicas irreversíveis sejam substituídas pelas portas reversíveis equivalentes.

O Computador Quântico - O bit quântico (q-bit)

- A Mecânica Quântica "mora" no Espaço de Hilbert que é um espaço vetorial dotado de um produto interno chamado norma.
- Agora teremos estados quânticos para representar os bits e portanto o chamaremos **q-bit**. Os valores 0 e 1 serão substituídos pelos **vetores** $|0\rangle$ e $|1\rangle$ representados por

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

- Grande diferença! Agora podemos ter um q-bit genérico $|\psi\rangle$ resultado da combinação linear dos estados $|0\rangle$ e $|1\rangle$

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

- α e β são números complexos.

O Computador Quântico - O bit quântico (q-bit)

- Os vetores $|0\rangle$ e $|1\rangle$ formam uma base ortonormal do espaço vetorial $\mathbb{C}^2$. Base computacional.
- $|\psi\rangle$ é chamado de **superposição** dos vetores da base com amplitudes α e β.
- $|\psi\rangle$ está simultaneamente nos estados $|0\rangle$ e $|1\rangle$!!!!
- Podemos armazenar uma quantidade de informação infinita em $|\psi\rangle$!! Essa informação está no mundo quântico.
- Para trazê la ao mundo clássico precisamos fazer uma medida. A medida altera o estado do q-bit!!
- Teremos então:
  - $|0\rangle$ com probabilidade $|\alpha|^2$;
  - $|1\rangle$ com probabilidade $|\beta|^2$.
- α e β não podem ser conhecidos através de uma medida. Além disso

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

O Computador Quântico - O bit quântico (q-bit)

- Podemos representar os q-bits geometricamente. Como $|\alpha|^2 + |\beta|^2 = 1$, podemos reescrever $|\psi\rangle$ usando coordenadas polares da seguinte forma

$$|\psi\rangle = e^{i\gamma}(\cos(\frac{\theta}{2})\,|0\rangle + e^{i\varphi}\sin(\frac{\theta}{2})\,|1\rangle) \tag{3}$$

onde θ,φ e γ são reais. O fator $e^{i\gamma}$ pode ser ignorado, pois ele não é observável e portanto

$$|\psi\rangle = \cos(\frac{\theta}{2})\,|0\rangle + e^{i\varphi}\sin(\frac{\theta}{2})\,|1\rangle \tag{4}$$

- Os ângulos θ e φ definem um ponto sobrea superfície de uma esfera de raio unitário.

O Computador Quântico - O bit quântico (q-bit)

- Nela podemos representar todos os estados de um q-bit

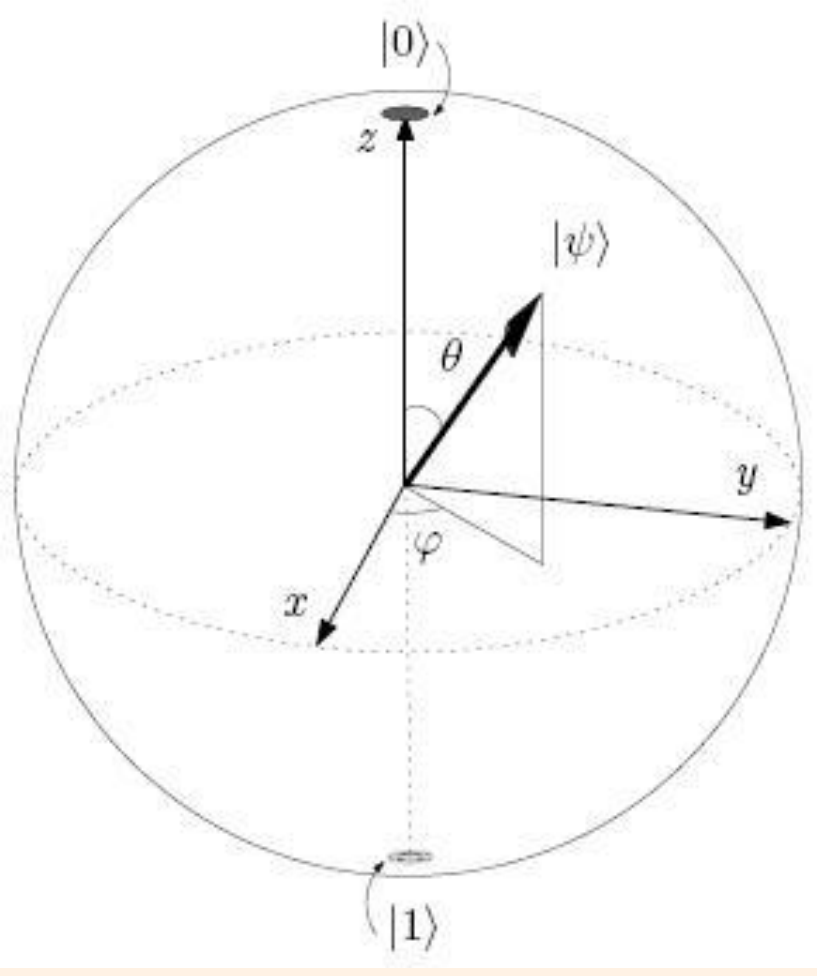


Figura: Representação de um q-bit na esfera de Bloch

O Computador Quântico - O bit quântico (q-bit)

- Não é possível calcular exatamente os valores de α e β mesmo que tenhamos um número grade de estados $|\psi\rangle$ iguais. Mesmo após repetidas medidas so teríamos os resultados $|0\rangle$ ou $|1\rangle$. Mesmo com a quantidade deles, isso só leva a um valor aproximado dos coeficientes pois são apenas probabilidades. Tanto sacrifício pra saber o valor de um único q-bit!!
- Paradoxal! Apesar de toda informação contida em um q-bit só temos acesso a 2 valores!! E agora?
- Temos ainda um outro fenômeno que ocorre com um estado quântico. A Mecânica Quântica nos diz que a evolução temporal de um sistema quântico isolado é descrita matematicamente por uma **transformação linear**, mais especificamente como estamos tratando de vetores unitários teremos então as transformações unitárias.

$$U^\dagger U = U U^\dagger = 1, \tag{5}$$

onde $U^\dagger = (U^*)^T$.

O Computador Quântico - O bit quântico (q-bit)

Há duas interações básicas de um computador quântico com os dados de entrada:

- Transformação unitária - evolução temporal atua no nível quântico, não temos acesso! Qualquer matriz unitária de ordem 2×2 pode ser usada! É um processo **reversível**.
- Medida. Faz a ligação entre o mundo quântico e clássico!
- Como fazer para aproveitarmos toda essa informação armazenada em um q-bit?

O Computador Quântico - Produto tensorial

Se quisermos tratar de sistemas de mais de um q-bit temos que introduzir o conceito de **produto tensorial**. Sejam dois estados

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_m \end{bmatrix} \quad \text{e} \quad |\varphi\rangle = \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_p \end{bmatrix}, \tag{6}$$

O produto tensorial entre eles resultará no vetor $|\chi\rangle = |\psi\rangle \otimes |\varphi\rangle$ com $mp-$linhas. Também podemos usar as seguintes notações para o produto tensorial $|\psi\rangle \otimes |\varphi\rangle$, $|\psi\rangle\,|\varphi\rangle$, $|\psi, \varphi\rangle$ e $|\psi\varphi\rangle$.

O Computador Quântico - Produto tensorial

$$|\chi\rangle = \begin{bmatrix} \psi_1\varphi_1 \\ \psi_1\varphi_2 \\ \vdots \\ \psi_1\varphi_p \\ \psi_2\varphi_1 \\ \psi_2\varphi_2 \\ \vdots \\ \psi_2\varphi_p \\ \vdots \\ \psi_m\varphi_1 \\ \psi_m\varphi_2 \\ \vdots \\ \psi_m\varphi_p \end{bmatrix}, \tag{7}$$

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

O Computador Quântico - Produto tensorial

Vejamos alguns exemplos:

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

e

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

O produto tensorial **não** é comutativo!

O Computador Quântico - Produto tensorial

Podemos generalizar o produto tensorial para matrizes quaisquer. Sejam as matrizes $A \in \mathbb{C}^{m \times n}$ e $B \in \mathbb{C}^{p \times q}$, a matriz $A \otimes B \in \mathbb{C}^{mp \times nq}$ é dada por:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}, \tag{8}$$

onde A_{ij} é o elemento da coluna i e linha j da matriz A.

O Computador Quântico - Produto tensorial

Exemplo:

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

então

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

O Computador Quântico - Produto tensorial

Vejamos algumas propriedades do produto tensorial. Sejam $z \in \mathbb{C}$, $v, v_1, v_2 \in \mathbb{C}^n$ e $w, w_1, w_2 \in \mathbb{C}^m$:

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

Dados dois operadores lineares A e B, podemos então definir um novo operador linear, $A \otimes B$, da seguinte forma

$$(A \otimes B)(|u\rangle \otimes |w\rangle) = A\,|u\rangle \otimes B\,|w\rangle\,. \tag{9}$$

Sejam $|\psi\rangle^{\otimes n}$ o produto tensorial de $|\psi\rangle$, por ele próprio n vezes e $A^{\otimes n}$ o produto dela própria n vezes.

O Computador Quântico - 2 q-bits

Seja $|\psi\rangle$ o estado genérico de 2 q-bits. Ele será uma superposição dos estados $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$, isto é,

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

onde

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

Podemos tentar simplificar a notação considerando os zeros e uns que aparecem nos vetores,$|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$ como números binários e escrevê- los em sua notação decimal

$$|0\rangle\,, |1\rangle\,, |2\rangle \quad \text{e} \quad |3\rangle\,.$$

O Computador Quântico - 2 q-bits

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

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \tag{11}$$

e as amplitudes α_i obedecendo à conservação de probabilidade

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1. \tag{12}$$

O Computador Quântico - Emaranhamento

Um estado de 2 q-bits pode ou não ser o resultado do produto tensorial de 2 estados de 1 q-bit! Sejam dois estados de 1 q-bits

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

e

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

onde a, b, c e d $\in \mathbb{C}$. Então

$$\begin{aligned} |\varphi\rangle \otimes |\psi\rangle &= (a\,|0\rangle + b\,|1\rangle) \otimes (c\,|0\rangle + d\,|1\rangle) \\ &= ac\,|00\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle\,. \end{aligned} \tag{13}$$

Temos então que um estado de 2 q-bits genérico (10)só é da forma (13)se, e somente se,

$$\alpha = ac,$$

$$\beta = ad,$$

$$\gamma = bc,$$

$$\delta = bd.$$

O Computador Quântico - Emaranhamento

Dessas relações de igualdade, temos que

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

Portanto,

$$\alpha\delta = \beta\gamma. \tag{14}$$

Em geral, um estado de 2 q-bits **não** é o produto tensorial de dois q-bits!! Um estado de 2 q-bits é dito estado **emaranhado** quando este estado não pode ser escrito como produto tensorial de dois estados de 1 q-bit.

O Computador Quântico - Emaranhamento

Vejamos:

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

Já o estado

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

é um estado **emaranhado**, pois não pode ser escrito como produto de dois q-bits.

Exercício: Prove que $|\zeta\rangle$ é um estado emaranhado!

O Computador Quântico - Produtos interno e externo

O produto interno entre dois estados $|\varphi\rangle$ e $|\psi\rangle \in \mathbb{C}^n$ é denotado por $\langle\varphi|\psi\rangle$ e definido como sendo o produto matricial entre $|\varphi\rangle^\dagger$ e $|\psi\rangle$, isto é,

$$\langle\varphi|\psi\rangle = (|\varphi\rangle)^\dagger |\psi\rangle = \sum_{i=1}^{n} \varphi_i^* \psi_i. \tag{15}$$

Propriedades:

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

com a,b $\in \mathbb{C}$ e $|\psi\rangle, |\varphi\rangle, |u\rangle, |v\rangle \in \mathbb{C}^n$.
Exercício: Demonstre as propriedades acima!

O Computador Quântico - Produtos interno e externo

A norma do estado $|\varphi\rangle$ é definida como sendo

$$\| |\varphi\rangle \| = \sqrt{\langle\psi|\psi\rangle}. \tag{16}$$

O produto externo entre dois estados $|\varphi\rangle \in \mathbb{C}^m$ e $|\psi\rangle \in \mathbb{C}^n$ é denotado por $|\varphi\rangle \langle\psi|$ e definido como sendo o produto matricial entre $|\varphi\rangle$ e $|\psi\rangle$,ou seja,

$$|\varphi\rangle \langle\psi| = |\varphi\rangle (|\psi\rangle)^\dagger. \tag{17}$$

O produto externo gera uma matriz de ordem $m \times n$.

O Computador Quântico - Produtos interno e externo

Sejam os estados de 1 q-bit

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

e

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

Temos, então,

$$\langle\varphi|\psi\rangle = \begin{bmatrix} a^* & b^* \end{bmatrix}\begin{bmatrix} c \\ d \end{bmatrix} = a^*c + b^*d,$$

$$\langle\varphi|\varphi\rangle = \begin{bmatrix} a^* & b^* \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = a^*a + b^*b,$$

já o produto externo,

$$|\varphi\rangle\langle\psi| = \begin{bmatrix} a \\ b \end{bmatrix}\begin{bmatrix} c^* & d^* \end{bmatrix} = \begin{bmatrix} ac^* & ad^* \\ bc^* & bd^* \end{bmatrix}.$$

O Computador Quântico

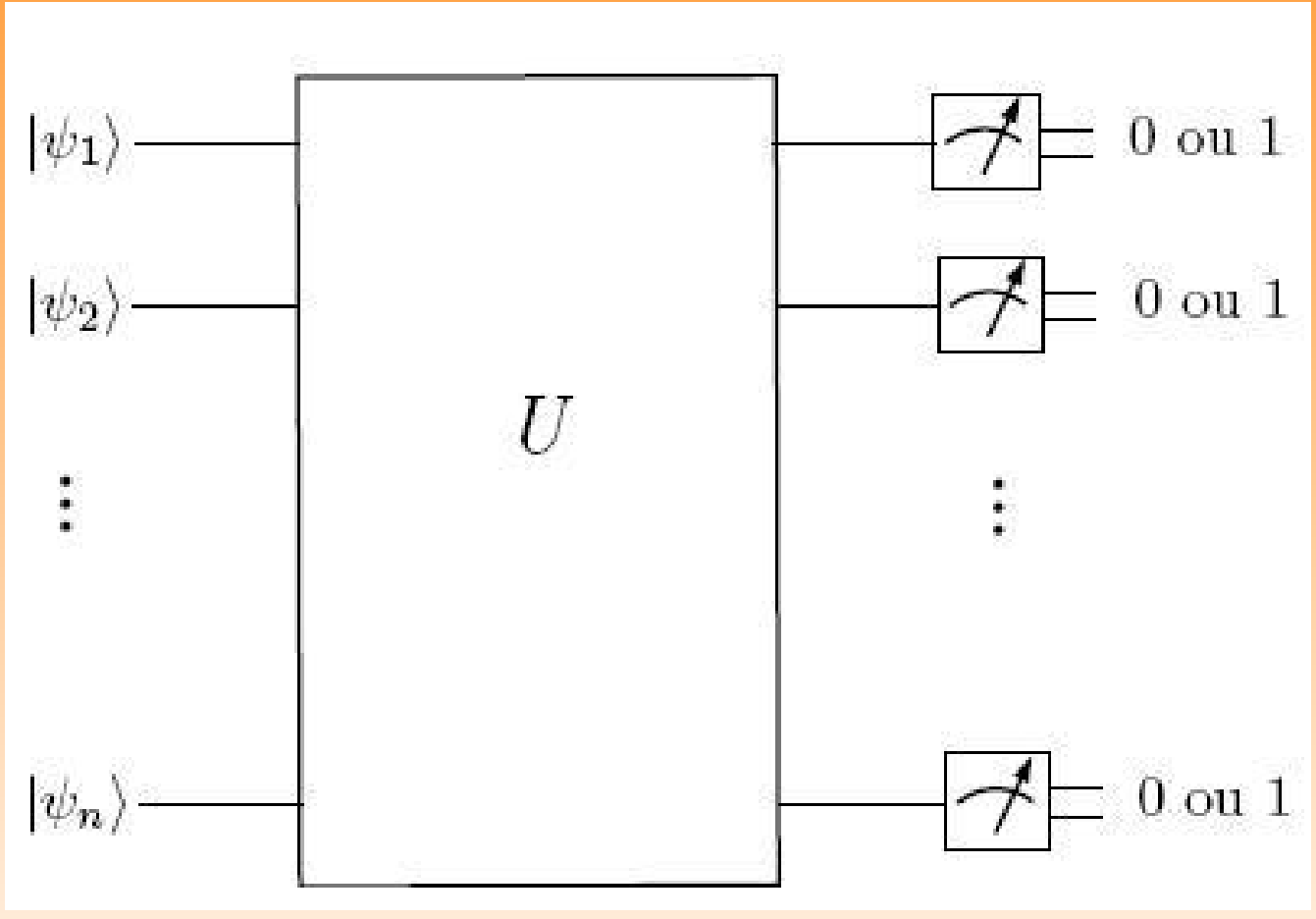


Figura: Esquema genérico para um computador quântico.

Circuitos Quânticos - Preliminares

A representação gráfica dos circuitos clássicos em geral, é bem próxima da sua implementação. Linhas correspodem aos fios e bifurcações nas linhas indicam que a corrente elétrica passa por ambos os fios. Vejamos que o que acontece nos circuitos quânticos é bem diferente.

Um circuito quântico é um dispositivo constituído de portas quânticas conectadas umas às outras e cujos passos computacionais são sincronizados no tempo. [8]

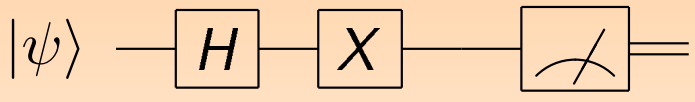


Figura: Circuito de 1 q-bit com duas portas unitárias e medida. Representando a operação $HX\,|\psi\rangle$.

Circuitos Quânticos- Notação e Convenções

- **Entrada**: pode ser qualquer estado emaranhado ou não;
- **Linhas horizontais**:Representam a evolução temporal de um q-bit;
- **Sentido**:o circuito é percorrido sempre da esquerda para a direita. Não há retroalimentação.

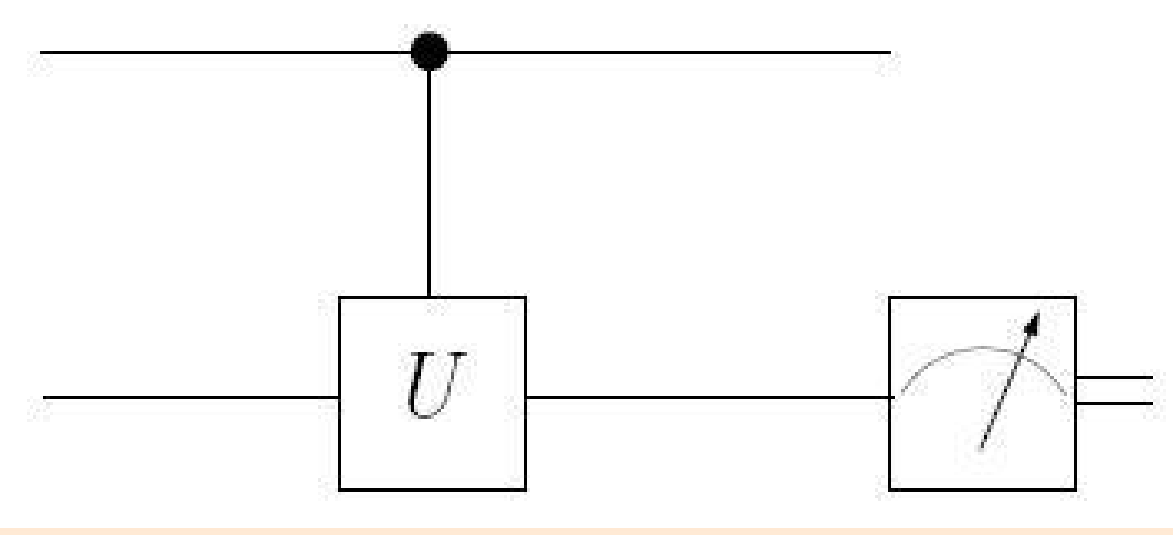


Figura: Porta quântica U-controlada.

Circuitos Quânticos- Notação e Convenções

- **Linhas verticais**: indicam que o circuito atua simultaneamente nos dois q-bits. Representa sincronismo, mas não há troca de informação.;
- **Controle**: o símbolo • indica que o q-bit desta mesma linha é um q-bit de controle. Se for $|0\rangle$ a porta U não atua, se for $|0\rangle$ a porta U é ativada;
- **Saída**: pode ou não haver medida. O q-bit inferior está sendo medido e o resultado será 0 ou 1.

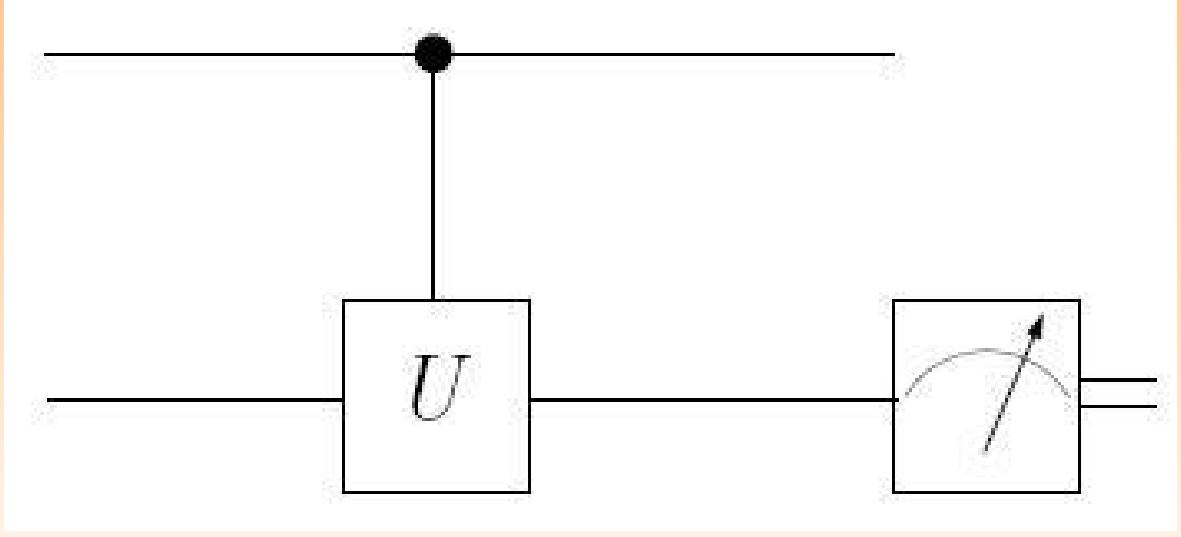


Figura: Porta quântica U-controlada.

Circuitos Quânticos - Porta NOT Quântica

- No caso clássico a porta NOT troca 0 por 1 e vice versa;
- No caso quântico temos o operador (transformação linear unitária) X
  - $X|0\rangle = |1\rangle$
  - $X|1\rangle = |0\rangle$
- Sua representação matricial é dada por

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

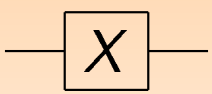


Figura: Porta X (NOT quântica).

Exercício: Prove que o operador X é unitário.

Circuitos Quânticos - Porta NOT Quântica

Situação sem contrapartida no caso clássico!

Seja o estado

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

aplicando o operador X a saída será

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

As probabilidades foram trocadas!!!

Circuitos Quânticos - Porta Hadamard

Esta é uma porta muito utilizada.

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

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

Figura: Porta Hadamard.

Exercício: Prove que a porta H é unitária.

Circuitos Quânticos - Porta Hadamard

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

Em notação decimal,

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

Generalizando para estados com n q-bits, obtemos:

$$\begin{aligned} H^{\otimes n}|0\ldots0\rangle &= (H|0\rangle)^{\otimes n} \\ &= \left(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\right)^{\otimes n} \\ &= \frac{1}{\sqrt{2^n}} \sum^{2^n-1} |i\rangle. \end{aligned}$$

Circuitos Quânticos - Porta de Fase ou Porta S

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad \text{ou} \quad \mathrm{S} = \begin{bmatrix} 1 & 0 \\ 0 & exp(i\pi/2) \end{bmatrix}$$

Aplicando o operador S a um estado genérico

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

a saída será

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

As probabilidades são as mesmas!!

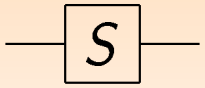


Figura: Porta de Fase.

Circuitos Quânticos - Porta $\pi/8$ Porta T

$$T = \begin{bmatrix} 1 & 0 \\ 0 & exp(i\pi/4) \end{bmatrix} \quad \text{ou} \quad \mathrm{T} = \exp(\mathrm{i}\pi/8)\begin{bmatrix} exp(-i\pi/8) & 0 \\ 0 & exp(i\pi/8) \end{bmatrix}$$

Aplicando o operador S a um estado genérico

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

obtemos

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

As probabilidades também serão as mesmas!!

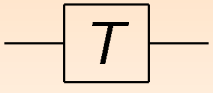
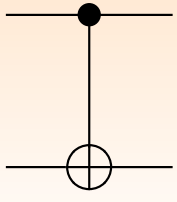


Figura: Porta T.

Circuitos Quânticos - Porta CNOT Quântica

Podemos usar as diversas portas de 1 q-bit para transformar o estado $|0\ldots\rangle$ de n q-bits em qualquer estado do tipo $|\psi_1\rangle\,|\psi_2\rangle\ldots|\psi_n\rangle$, onde cada um desses $|\psi_i\rangle$ é uma superposição arbitrária $\alpha\,|0\rangle + \beta\,|1\rangle$. Mas todos estes estados são do tipo produto, ou seja, não emaranhados. Para obtermos estados emaranhados precisamos de portas que atuem sobre os múltiplos q-bits.

- 2 q-bits de entrada **controle** e **alvo**;
- Se o bit de controle está no estado $|1\rangle$ ela é ativada caso contrário não;
- Os bits alvo e controle podem estar superpostos ou emaranhados.
- A porta CNOT é universal. Qualquer operador unitário pode ser representado usando portas CNOT e portas de um q-bit.



$$\begin{aligned} |00\rangle &\rightarrow |00\rangle \\ |01\rangle &\rightarrow |01\rangle \\ |10\rangle &\rightarrow |11\rangle \\ |11\rangle &\rightarrow |10\rangle \end{aligned}$$

Circuitos Quânticos - Porta Toffoli Quântica

Esta é uma porta controlada por dois q-bits

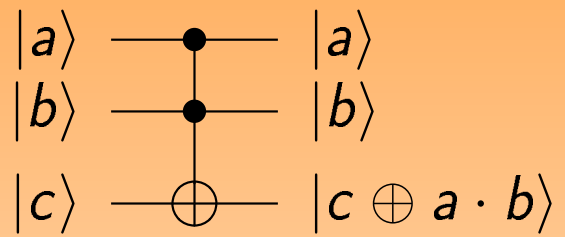


Figura: Porta Toffoli Quântica.

Sua ação na base computacional pode ser representada por:

$$|i, j, k\rangle \rightarrow |i, j, k \oplus ij\rangle \,, \tag{18}$$

onde $i, j \in \{0, 1\}$ e $\oplus$ é a adição módulo 2. A base computacional possui 8 elementos. Em geral ela é muito utilizada para simplificar a representação de circuitos quânticos.

Circuitos Quânticos - Porta Toffoli Quântica

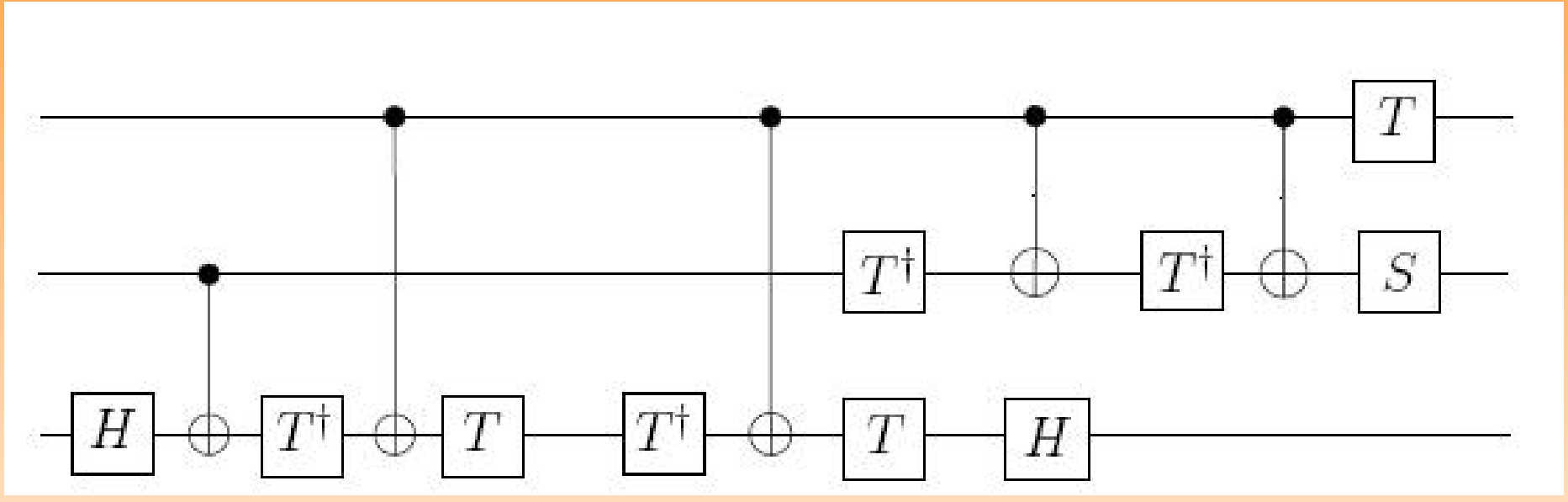


Figura: Decomposição da porta Toffoli em portas de 1 q-bit e portas CNOT.

Circuitos Quânticos - Paralelismo Quântico

Uma operação quântica é sempre unitária e portanto reversível. Então, um computador quântico precisa de dois registradores para realizar uma computação:

- Um registrador para guardar o estado de entrada;
- Um registrador para guardar o estado de saída.

A computação de uma função f é determinada por uma operação unitária U_f que deve atuar sobre os dois registradores preservando a entrada e efetuando a operação no segundo.

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle . \tag{19}$$

Se $y = 0$, então,

$$U_f |x\rangle |0\rangle = |x\rangle |0 \oplus f(x)\rangle = |x\rangle |f(x)\rangle . \tag{20}$$

Circuitos Quânticos - Paralelismo Quântico

Suponha agora que preparamos um registrador com m q-bits no estado $|\psi\rangle$ de superposição igualmente distribuída e um registrador com n q-bits no estado $|0\rangle$

$$|\psi\rangle\,|0\rangle = \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle\,|0\rangle.$$

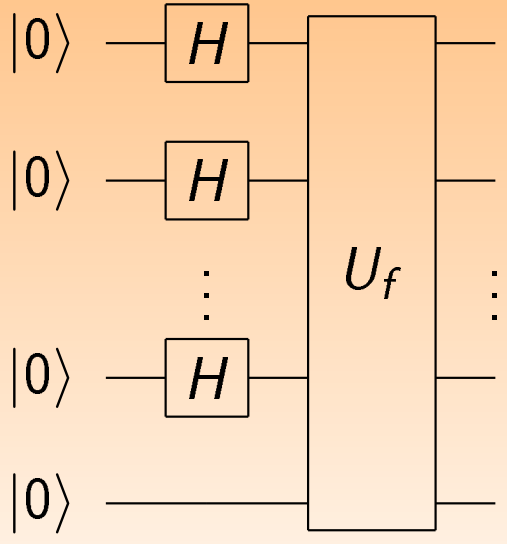


Figura: Circuito que calcula o valor de f para todos os valores de x.

Circuitos Quânticos - Paralelismo Quântico

Aplicando U_f a este estado obtemos:

$$\begin{aligned} U_f |\psi\rangle |0\rangle &= U_f \left(\frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |0\rangle \right) \\ &= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} U_f |x\rangle |0\rangle \\ &= \frac{1}{\sqrt{2^m}} \sum_{x=0}^{2^m-1} |x\rangle |f(x)\rangle. \end{aligned} \tag{21}$$

Este circuito realiza o cálculo de todos os 2^m valores $f(0), f(1), \ldots, f(2^m - 1)$ ao mesmo tempo com uma única aplicação da operação unitária U_f. Este é o **Paralelismo Quântico**.

O problema de Grover

O problema de Grover

O problema de Grover é o de encontrar um determinado elemento em uma lista desordenada contendo N elementos.

Classicamente, no pior caso, teríamos que testar todos os N elementos da lista. Usando as propriedades da Mecânica Quântica , a quantidade de "testes" necessários para identificar o elemento procurado é proporcional a $\sqrt{N}$[9, 10]. Podemos representar matematicamente este problema, seja a busca realizada sobre uma lista $\{0, 1, \ldots, N-1\}$, onde $N = 2^n$ e que a função f utilizada para reconhecer o elemento procurado i_0 seja definida por:

$$f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}, \tag{22}$$

e

$$f(i) = \begin{cases} 1, & \text{se} \quad i = i_0 \ , \\ 0, & \text{se} \quad i \neq i_0 \ , \end{cases} \tag{23}$$

Operadores do algoritmo

- O algoritmo de Grover utiliza 2 registradores quânticos:
  - O primeiro com n q-bits, inicializado no estado $|0\ldots0\rangle$. Está relacionado aos elementos da lista;
  - O segundo com 1 q-bit, inicializado no estado $|1\rangle$.
- A cada elemento i da lista $\{0, 1, \ldots, N-1\}$, associaremos o estado $|i\rangle$ de n q-bits.

Operadores do algoritmo

Antes mesmo da execução propriamente dita do algoritmo, o primeiro registrador é alterado para formar a superposição de todos os estados associados aos elementos da lista aplicando - se o operador de Hadamard em cada um dos q-bits do 1^o registrador.

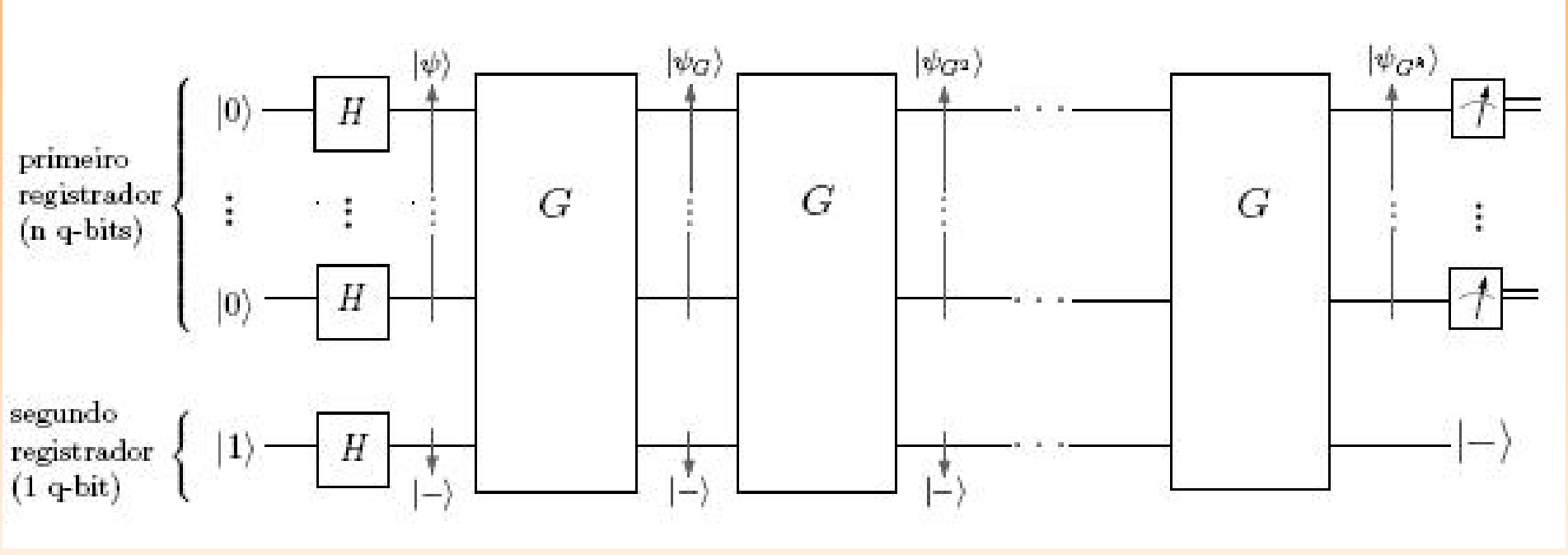


Figura: Esquema genérico para o algoritmo de Grover.

Operadores do algoritmo

Temos então a superposição de $N = 2^n$ estados (elementos da lista) com a mesma amplitude.

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle \tag{24}$$

Também aplicamos o operador de Hadamard ao segundo registrador e obtemos o estado $|-\rangle$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{25}$$

Operadores do algoritmo

Precisamos de um operador linear unitário para representar quanticamente a função f que utilizamos para a identificação do elemento procurado. Seja o operador U_f

$$|i\rangle\,|0\rangle \xrightarrow{U_f} |i\rangle\,|f(i)\rangle \tag{26}$$

Então usando este operador temos:

$$U_f(|i\rangle\,|0\rangle) = \begin{cases} |i\rangle\,|1\rangle\,, & \text{se} \quad i = i_0\ , \\ |i\rangle\,|0\rangle\,, & \text{se} \quad i \neq i_0\ , \end{cases} \tag{27}$$

Para completar a definição, definimos também

$$U_f(|i\rangle\,|1\rangle) = \begin{cases} |i\rangle\,|0\rangle\,, & \text{se} \quad i = i_0\ , \\ |i\rangle\,|1\rangle\,, & \text{se} \quad i \neq i_0\ , \end{cases} \tag{28}$$

Só altera o segundo registrador quando o primeiro tem o elemento procurado!

Operadores do algoritmo

O operador U_f que atua no estado todo (produto tensorial)está bem definido pois basta definir sua atuação nos elementos da base.
Podemos representar u_f da seguinte forma:

$$U_f(|i\rangle\,|j\rangle) = |i\rangle\,|j \oplus f(i\rangle\,, \tag{29}$$

onde $|i\rangle$ é o estado de n q-bits do primeiro registrador
($i \in \{0, 1, \ldots, N-1\}$), $|j\rangle$ é o estado de 1 q-bit do segundo registrador.

- U_f simula quanticamente o papel da função f;
- Para identificar o elemento procurado i_0 bastaria aplicar U_f em cada estado associado aos elementos da lista e manter o segundo registrador no estado $|0\rangle$ ou $|1\rangle$.
- Mas isso não traria ganho algum em relação ao caso clássico.
- Precisamos aplicar U_f nos estados superpostos!

Operadores do algoritmo

Vamos então aplicar U_f no estado resultante da inicialização

$$U_f \left|\psi\right\rangle \left|-\right\rangle = U_f \left(\left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left|i\right\rangle \right) \left|-\right\rangle \right) \tag{30}$$

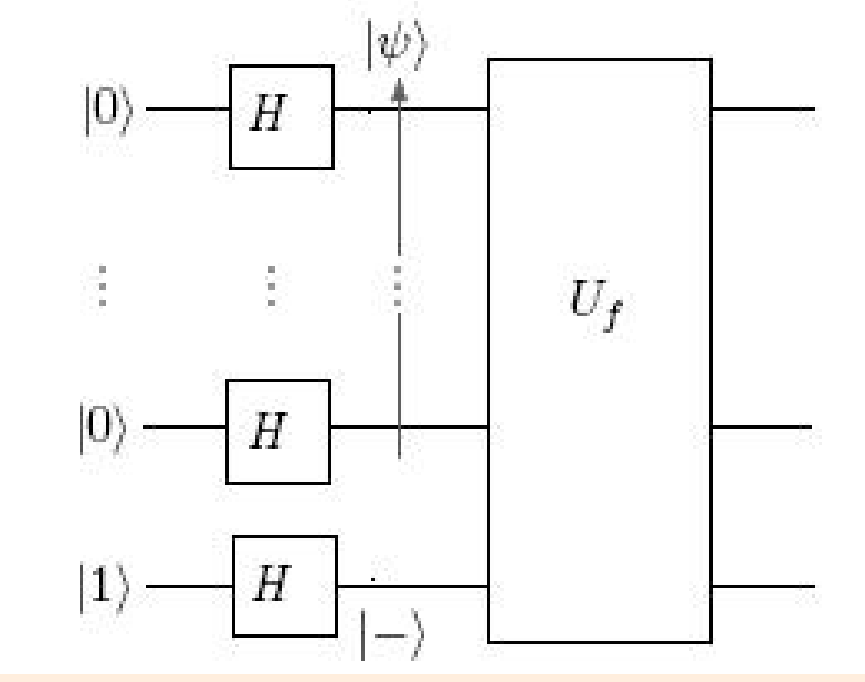


Figura: Aplicação do operador U_f sobre o estado $\left|\psi\right\rangle \left|-\right\rangle$.

Operadores do algoritmo

Então, obtemos

$$\begin{aligned} U_f |\psi\rangle |-\rangle &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} U_f \left(\left(|i\rangle \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \right) \right) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} U_f \left(\frac{1}{\sqrt{2}} (|i\rangle |0\rangle - |i\rangle |1\rangle) \right) \end{aligned} \tag{31}$$

E da linearidade de U_f,

$$U_f |\psi\rangle |-\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2}} (U_f |i\rangle |0\rangle - U_f |i\rangle |1\rangle) \tag{32}$$

Operadores do algoritmo

Então, obtemos da definição de U_f, finalmente

$$\begin{aligned} U_f |\psi\rangle |-\rangle &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2}} (|i\rangle |f(i)\rangle - |i\rangle |1 \oplus f(i)\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \frac{1}{\sqrt{2}} |i\rangle (|f(i)\rangle - |1 \oplus f(i)\rangle) \end{aligned} \tag{33}$$

E da definição de f temos

$$|i\rangle (|f(i)\rangle - |1 \oplus f(i)\rangle) = \begin{cases} |i\rangle (|1\rangle - |0\rangle), & \text{se } i = i_0 \ , \\ |i\rangle (|0\rangle - |1\rangle), & \text{se } i \neq i_0 \ , \end{cases} \tag{34}$$

Operadores do algoritmo

Substituindo então temos

$$U_f |\psi\rangle |-\rangle = \frac{1}{\sqrt{2^n}} \left(\sum_{i=0}^{2^n-1} \left(\frac{1}{\sqrt{2}} (|i\rangle (|0\rangle - |1\rangle) \right) + (|i\rangle (|1\rangle - |0\rangle) \right) \quad (35)$$

E então da definição de $|-\rangle$

$$\begin{aligned} U_f |\psi\rangle |-\rangle &= \frac{1}{\sqrt{2^n}} \left(\left(\sum_{i=0, i\neq i_0}^{2^n-1} (|i\rangle |-\rangle) \right) - |i_0\rangle |-\rangle) \right) \\ &= \left(\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} (-1)^{f(i)} |i\rangle \right) |-\rangle \quad (36) \end{aligned}$$

O estado do segundo registrador não se altera!! Usamos o paralelismo quântico. A amplitude do elemento procurado teve seu sinal alterado.

Operadores do algoritmo

Dessa forma o elemento procurado foi marcado lá no Mundo quântico. Precisamos agora aumentar a probabilidade desse elemento ser obtido após uma medida. O novo estado do primeiro registrador será $|\psi_1\rangle$, isto é,

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{f(i)} |i\rangle \tag{37}$$

Uma representação geométrica para esse novo estado $|\psi_1\rangle$ é como sendo uma reflexão desse vetor em relação ao subespaço ortogonal ao $|i_0\rangle$, gerado por todos os outros elementos da base computacional. Suponha que nosso estado possa ser representado como sendo gerado pelos vetores ortogonais $|u\rangle$(vetor das não soluções) e $|i_0\rangle$(vetor da solução), então

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i \neq i_0}^{N-1} |i\rangle + \frac{1}{\sqrt{N}} |i_0\rangle \,, \tag{38}$$

e então

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum^{N-1} |i\rangle - \frac{1}{\sqrt{N}} |i_0\rangle \tag{39}$$

Operadores do algoritmo

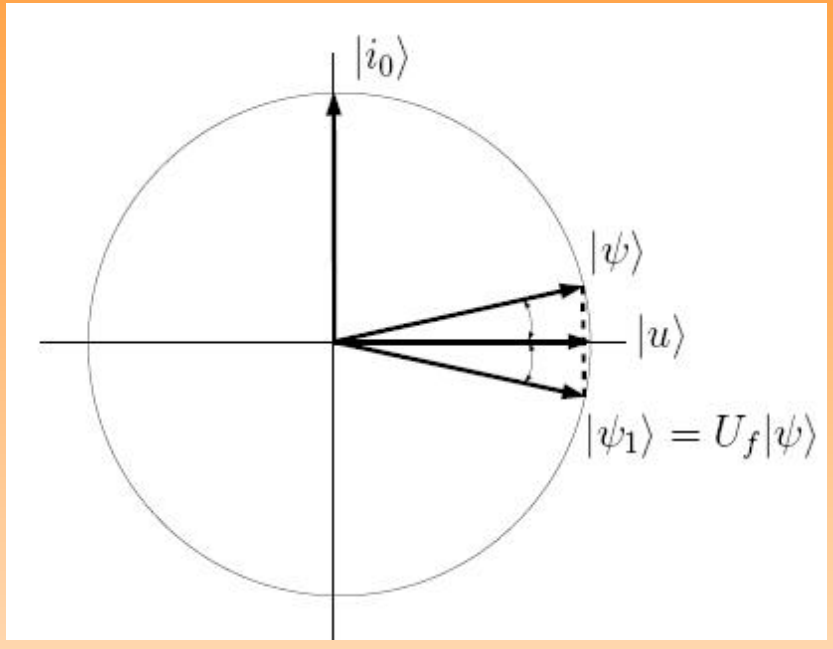


Figura: Ação de U_f no estado $|\psi\rangle$

Essa reprsentação nos induz então a refletir o vetor $|\psi_1\rangle$ em relação vetor $|\psi\rangle$ para aumentar a amplitude do elemento procurado $|i_0\rangle$ em relação à sua amplitude no estado $|\psi\rangle$

Operadores do algoritmo

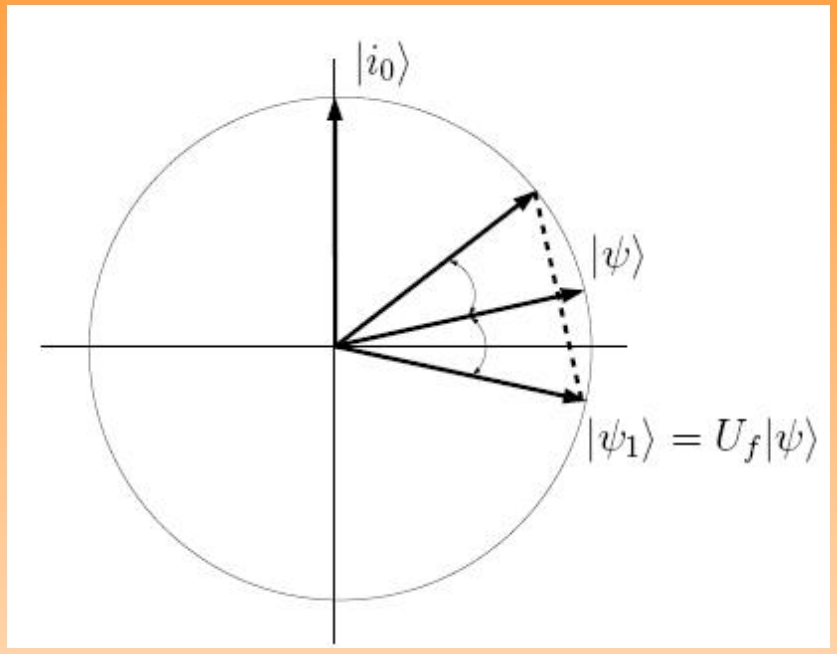


Figura: Reflexão de $|\psi_1\rangle$ em relação a$|\psi_1\rangle$

Essa reflexão ou amplificação de fase é feita com o seguinte operador unitário

$$2\,|\psi\rangle\,\langle\psi| - I, \tag{40}$$

onde I é o operador identidade.

Operadores do algoritmo

$$
\begin{aligned}
|\psi_G\rangle &= (2\,|\psi\rangle\langle\psi| - I)\,|\psi_1\rangle \\
&= (2\,|\psi\rangle\langle\psi| - I)\left(|\psi\rangle - \frac{2}{\sqrt{N}}\,|i_0\rangle\right) \\
&= \frac{N-4}{N}\,|\psi\rangle + \frac{2}{\sqrt{N}}\,|i_0\rangle\,. \qquad (41)
\end{aligned}
$$

Esse é o estado do primeiro registrador após a aplicação dos operadores U_f e $2\,|\psi\rangle\langle\psi| - I$. A composição desses dois operadores é conhecida como *operador de grover* G, isto é,

$$G = ((2\,|\psi\rangle\langle\psi| - I) \otimes I)U_f, \qquad (42)$$

Essa combinação dessas duas reflexões (aplicação do operador G) leva a uma rotação do estado inicial de θ radianos.

Operadores do algoritmo

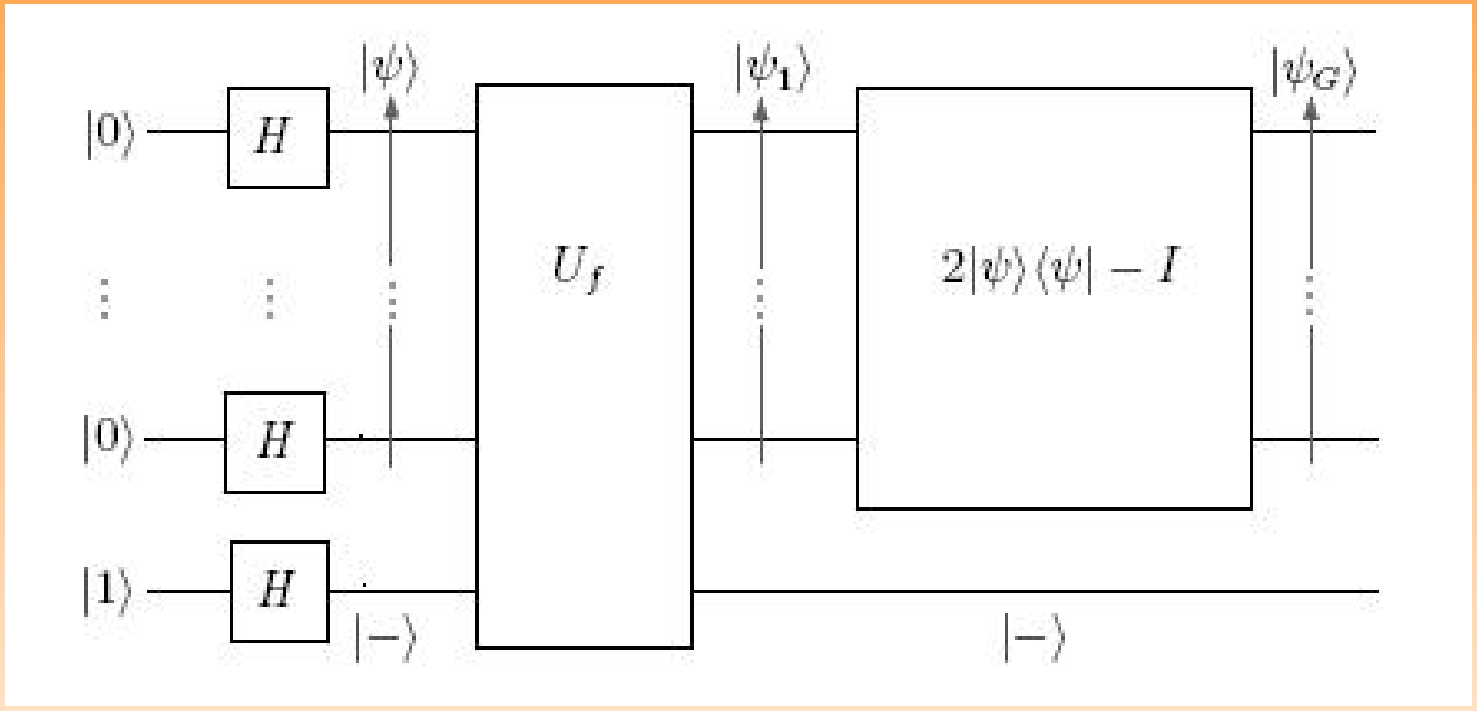


Figura: Uma aplicação do operador de Grover

Operadores do algoritmo

Então temos que o estado $|\psi_G\rangle$ pode ser escrito como,

$$\begin{aligned} |\psi_G\rangle &= \frac{N-4}{N}|\psi\rangle + \frac{2}{\sqrt{N}}|i_0\rangle \\ &= \frac{N-4}{N}\left(\frac{1}{\sqrt{N}}\sum_{i\neq i_0}^{N-1}|i\rangle + \frac{1}{\sqrt{N}}|i_0\rangle\right) + \frac{2}{\sqrt{N}}|i_0\rangle\,. \end{aligned} \tag{43}$$

E portanto, a amplitude do estado $|i_0\rangle$, após a primeira aplicação do operador G é

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

Operadores do algoritmo

Para $N = 4$

- Medindo o estado $|\psi_G\rangle$ encontraremos o estado $|i_0\rangle$ com 100% de acerto!!
- Já se medirmos o o estado $|\psi\rangle$ obteremos $|i_0\rangle$ com 25%!

Para valores grandes de N, essa probabilidade ainda é pequena!

Operadores do algoritmo

Cada aplicação do operador G aumenta a amplitude do estado $|i_0\rangle$ em relação à sua amplitude no estado $|\psi\rangle$. Se quisermos aumentar ainda mais essa amplitude devemos aplicar o operador de Grover repetidas vezes.

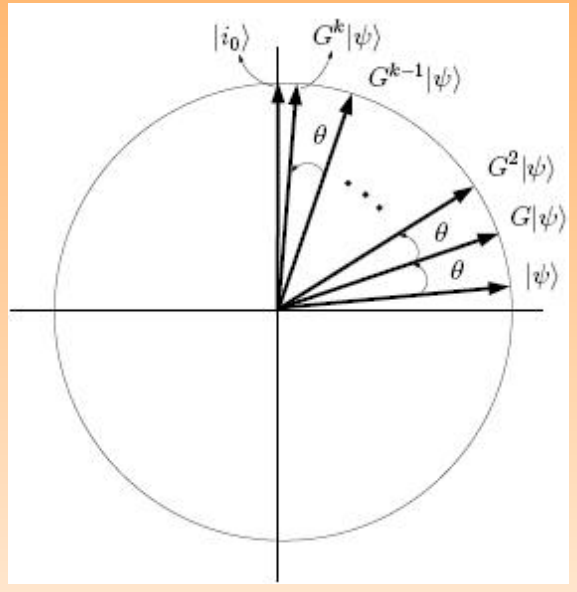


Figura: Aplicações sucessivas do operador de Grover

Operadores do algoritmo

O número de de vezes que é necessário aplicar o operador de Grover é dado por:

$$R \leq \lceil \frac{\pi}{4} \sqrt{N} \rceil \tag{44}$$

Dessa equação vemos que a complexidade computacional do algoritmo de Grover é da da ordem de $O\sqrt{N}$.

Ganho quadrático em relação ao caso clássico!!

Algoritmo de Shor

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer (1997)

Peter W. Shor

SIAM Journal on Computing

- Marco no desenvolvimento da computação quântica!
- Primeiro exemplo de um algoritmo quântico com ganho exponencial em relação aos algoritmos clássicos conhecidos.
- Tem aplicação prática relevante.
- A principal é a quebra dos métodos de criptografia mais usados.
- Também é possível calcular o logaritmo discreto.
- É paradigma para a solução do "Problema do Subgrupo Escondido"

Algoritmo de Shor

- Objetivo: encontrar os fatores primos de um número composto N

$$N \stackrel{?}{=} P \times Q$$

$$15 \stackrel{?}{=} 5 \times 3$$

Imagine que N é um número grande, por exemplo com 300 dígitos na notação decimal, já que tais números são usados em criptografia. Embora N seja grande, o número de q-bits necessário para guardá-lo é muito pequeno. Em geral, $n = \log_2 N$ não é um inteiro, então definimos

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

Algoritmo de Shor

- Um computador quântico com n q-bits pode guardar N ou qualquer outro inteiro positivo menor que N. O número de fatores primos de N é no máximo n.
- Se o número de q-bits e o número de fatores primos são menores ou iguais a n, então é natural perguntar se existe um algoritmo que fatora N em um número de passos que é polinomial em n. Mais precisamente a questão é: existe um algoritmo de fatoração na classe de complexidade P? [11]

Algoritmo de Shor

- Fatoração: reduz-se ao cálculo de achar a ordem um inteiro x módulo N. Se x e N possuem fatores comuns, então o $MDC(x, N)$ fornece um fator de N, portanto é suficiente investigar o caso quando x é coprimo com N.
- A ordem de x é o menor inteiro r tal que

$$x^r \equiv 1 \text{mod } N$$

  x é escolhido aleatoriamente.
- Números coprimos são os que não têm nenhum fator comum maior do que 1.

Algoritmo de Shor

Theorem

Se r for par, então $(x^{\frac{r}{2}} \pm 1)$ contém fatores comuns com N.

- Exemplo: $N = 21$ e $x = 2$

$$2^4 \equiv 16 \mathrm{mod} 21$$

$$2^5 \equiv 11 \mathrm{mod} 21$$

$$2^6 \equiv 1 \mathrm{mod} 21$$

A ordem de 2 módulo 21, é $r = 6$. Então $2^3 \pm 1$ tem fatores de $N = 21$. Com efeito, $2^3 + 1 = 9$ tem fator 3, $2^3 - 1 = 7$ tem fator 7.

Algoritmo de Shor

- Escolha aleatoriamente um inteiro x menor que N
- Ache a ordem de x módulo N
- Se a ordem não for par, rode o algoritmo novamente
- Como determinar a ordem de x classicamente?
  Calcule

$$x^1, x^2, x^3, \ldots$$

O circuito de Shor

O circuito que implementa o algoritmo de Shor possui dois registradores:

- O primeiro registrador possui t q-bits, onde t deve ser escolhido tal que $N^2 \leq 2^t \leq 2N^2$ para que a probabilidade de achar o resultado seja maior que $\frac{1}{2}$.
- O segundo registrador possui n q-bits, onde $n = \lceil \log_2 N \rceil$.
- Verifica-se que se a ordem que estamos procurando é uma potência de 2 então é suficiente tomar $t = n$.

Algoritmo de Shor

1. Inicialize o computador quântico no estado

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

2. Aplique $H^{\otimes n}$ ao 1° registrador ($N \leq 2^n$):

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle\,|0\rangle$$

Algoritmo de Shor

1. Aplique o operador $V_x(|j\rangle |k\rangle) = |j\rangle \left|k + x^j\right\rangle$

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

As operações são feitas módulo N. Gera todas as potências de x simultaneamente!

Algoritmo de Shor

Quando j for um múltiplo de r:

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

O estado $|\psi_2\rangle$ pode ser reescrito como:

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

O primeiro registrador é periódico em r! A informação que nos interessa é um período. Qualquer resultado x^0,x^1, ...,x^{r-1} pode ser obtido com igual probabilidade.

Algoritmo de Shor

1. Meça o 2° registrador. O resultado da medida será:

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

Note que depois da medida, a constante é renormalizada para $\sqrt{\frac{r}{2^n}}$, pois restaram $\frac{2^n}{r}$ termos na soma.

Algoritmo de Shor - Distribuição de probabilidade

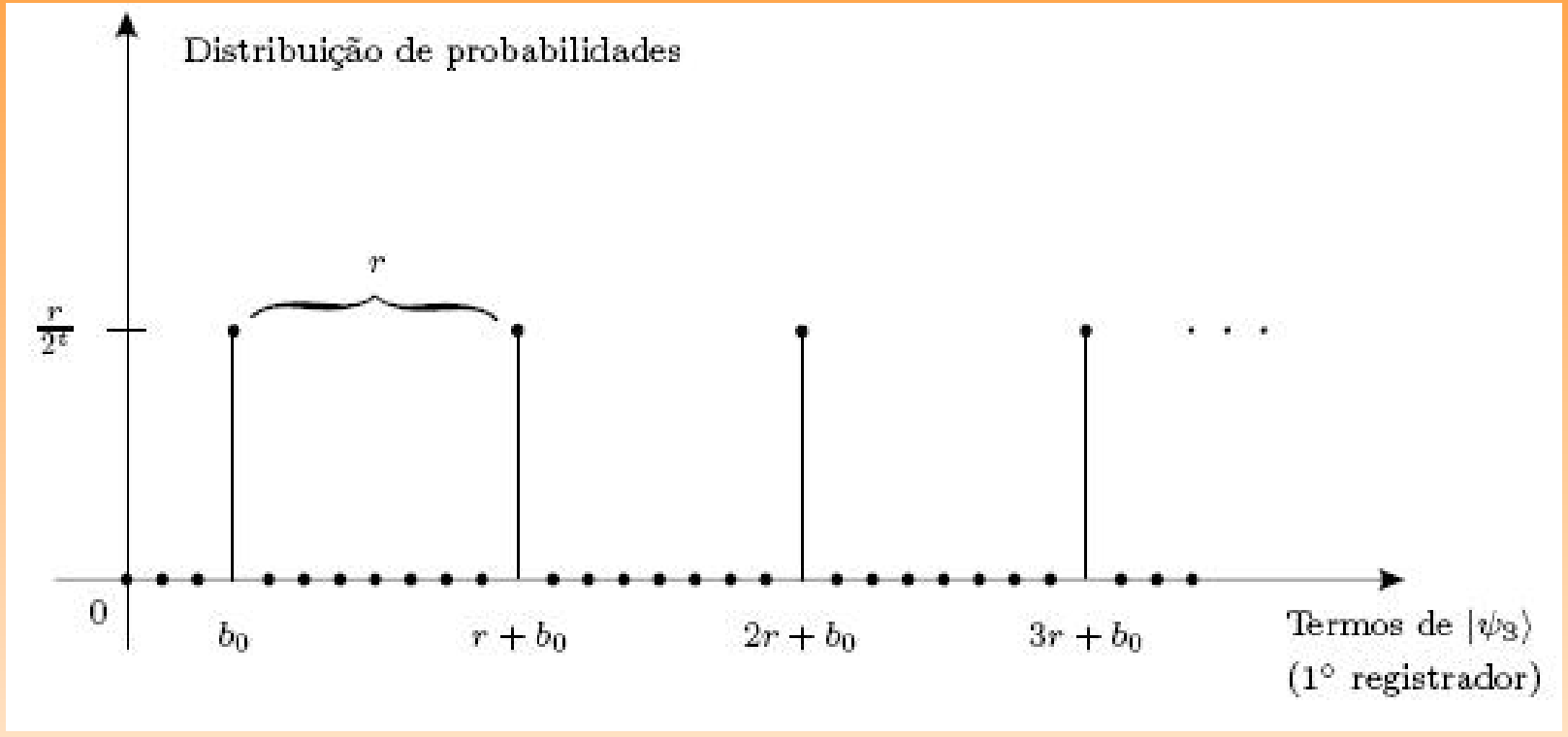


Figura: Distrib. de probab. de $|\psi_3\rangle$ com $b_0 = 3$ e $r = 8$.

A probabilidade forma uma função periódica de período r.

O Algoritmo Quântico

Como podemos descobrir o período de uma função eficientemente?

- A resposta é a transformada de Fourier.
- A transformada de Fourier de uma função periódica de período r é uma nova função com período proporcional a $1/r$.
- Todo o método depende de um algoritmo quântico eficiente para se calcular a transformada de Fourier. Algo que não existe classicamente.

O Algoritmo Quântico

Como podemos descobrir o período de uma função eficientemente?

- A resposta é a transformada de Fourier.
- A transformada de Fourier de uma função periódica de período r é uma nova função com período proporcional a $1/r$.
- Todo o método depende de um algoritmo quântico eficiente para se calcular a transformada de Fourier. Algo que não existe classicamente.

Transformada de Fourier Quântica

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

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

TF de um estado da base computacional

$$|\psi_j\rangle \equiv \mathrm{TF}_N(|j\rangle) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi i j k}{N}} |k\rangle$$

O Algoritmo Quântico

1. Aplique $\mathrm{TF}_{2^n}^{\dagger}$ ao estado $|\psi_3\rangle$:

$$|\psi_4\rangle = \sqrt{\frac{r}{2^n}} \sum_{a=0}^{\frac{2^n}{r}-1} \left(\frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} e^{-2\pi i j(ar+b_0)/2^n} |j\rangle \right) \left| x^{b_0} \right\rangle .$$

O Algoritmo Quântico

Invertendo a ordem do somatório, temos

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

A expressão no colchetes é igual a 1, quando $j = k\frac{2^n}{r}$, $k = 0, \ldots, r-1$ e 0 caso contrário. Então

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

O Algoritmo Quântico

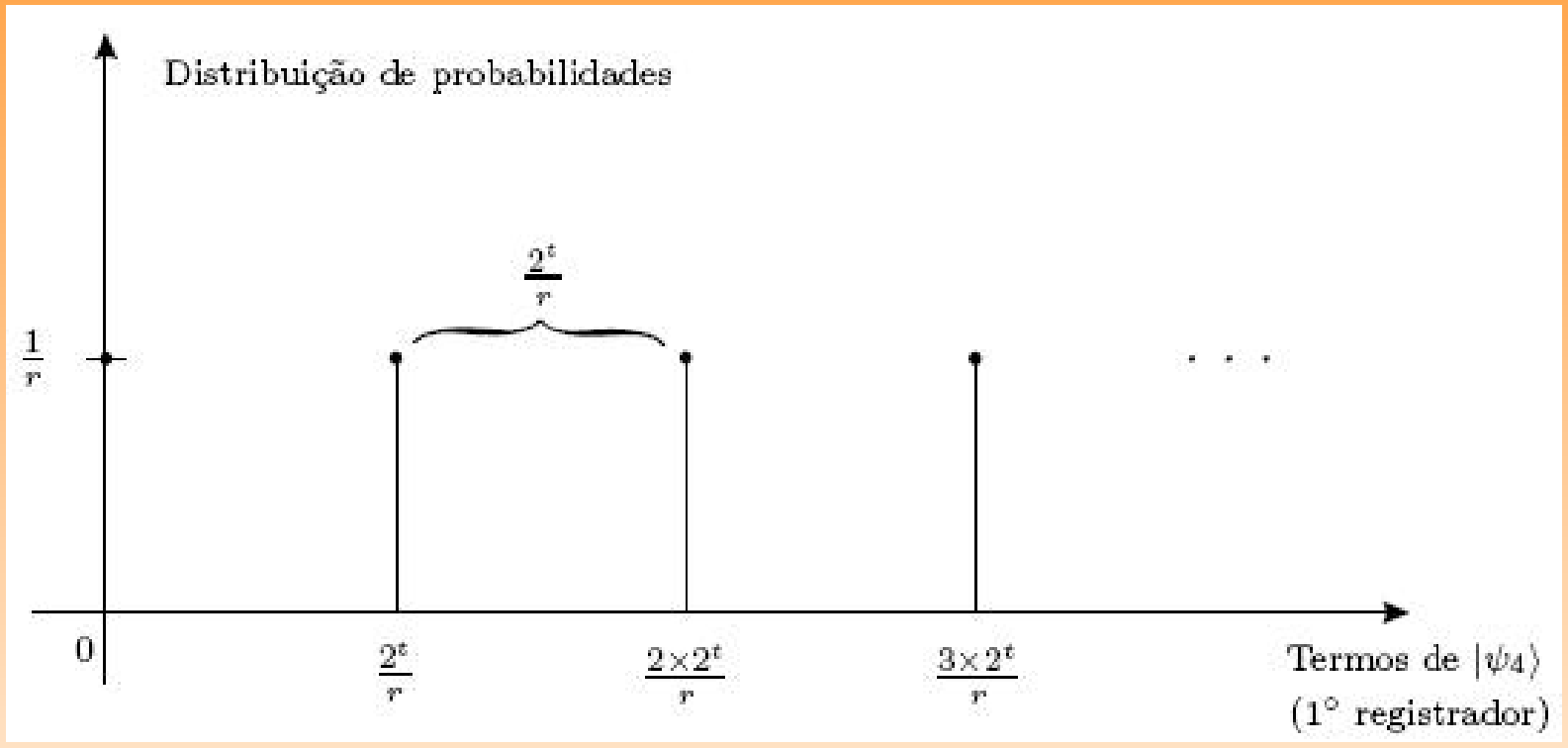


Figura: Distrib. de probab. de $|\psi_4\rangle$ com $b_0 = 3$ e $r = 8$.O número de picos é r e o período é $2^t/r$.

Passo final do algoritmo

- Meça o primeiro registrador.
- Divida $\left|\frac{k2^n}{r}\right\rangle$ por 2^n e obtenha $\frac{k}{r}$, com $0 \leq k \leq r-1$.
- Se k e r n ao tiverem fatores comuns e $k \neq 0$, teremos r na primeira tentativa, pois basta tomar o denominador.
- Caso contrário, temos que rodar a parte quântica do algoritmo novamente. Com poucas repetições se chega ao resultado.
- A complexidade é $O(n^2)$.

Circuito do Algoritmo de Shor

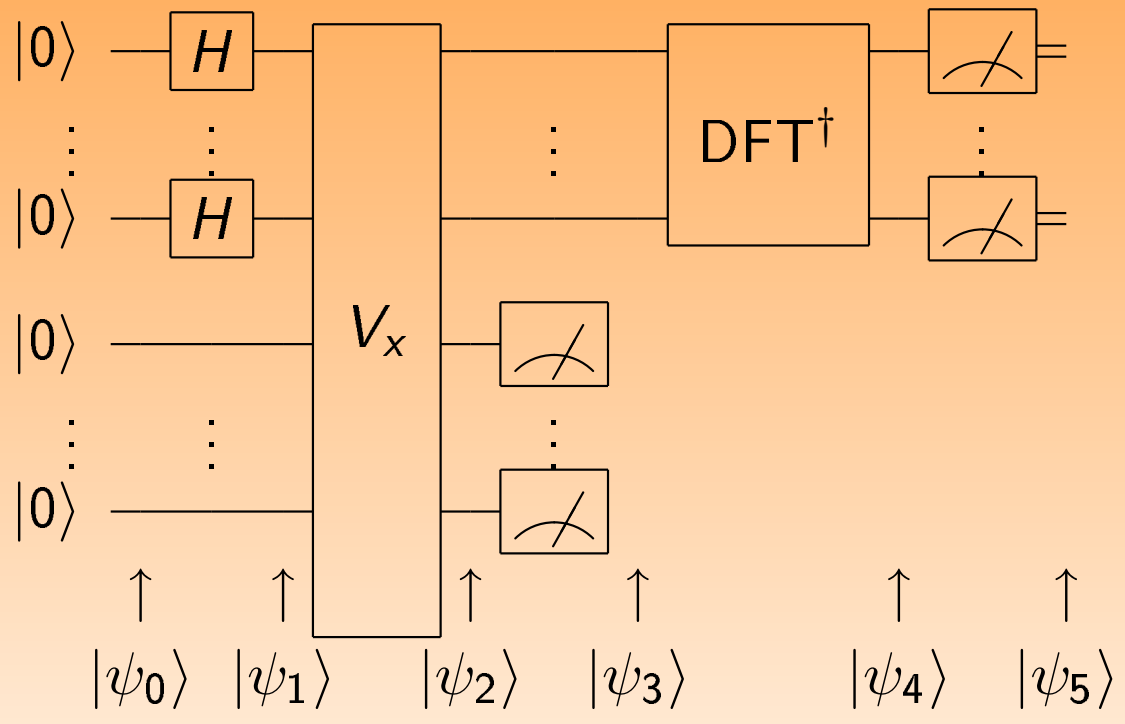


Figura: Circuito resolvedor para o algoritmo de Shor.

Caminhos aleatórios clássicos

- Em 1827 Robert Brown observa o movimento errático de grãos de pólen sobre a água.
- Caminhos aleatórios clássicos são aplicados em muitas áreas do conhecimento.
- São usados para modelar o comportamento difusivo de moléculas em líquidos e gases.
- São aplicados em economia, genética populacional, física, psicologia e diversas outras áreas.
- Em Ciência da Computação eles são peça fundamental devido ao seu uso no desenvolvimento de algoritmos estocásticos

Caminhos aleatórios clássicos

Eles são usados em Ciência da Computação teórica e aplicada estão na base de vários algoritmos clássicos.

- Estimativa do volume de um corpo convexo [12];
- Aproximação do permanente de uma matriz [13];
- 3-SAT [14];
- Conectividade de grafos [15];
- Simulações de Monte Carlo.

Esses são grandes motivos pra se acreditar que estudar o caminho aleatório quântico deva ser um caminho natural na busca de algoritmos quânticos.

Caminhantes Quânticos

- Em 1993, Y.Aharonov,L. Davidovich e N. Zagury usam pela primeira vez o termo caminho aleatório quântico. E já apresentam algumas das peculiaridades do comportamento quântico [16].
  A partir deste trabalho, surgem dois modelos de caminhos quânticos:
- Em 1998 o modelo contínuo no tempo devido a Farhi e Gutmann [17];
- Em 2001 o modelo discreto no tempo devido a Aharanov, Ambainis, Kempe e Vazirani em 2001 [18]. Ganha um aspecto mais formal e aplicável como possível ferramenta computacional.

Após esse artigo essa nova área de pesquisa desponta e surgem diversos trabalhos nas mais variadas direções.

Caminho na reta - caminho aleatório clássico

1. Inicie sua partícula na origem: $x = 0$
2. Lance a moeda
3. Mova a partícula uma posição para esquerda ou direita de acordo com o resultado da moeda
   - Cara : $x \longrightarrow x + 1$
   - Coroa: $x \longrightarrow x - 1$
4. Repetir t vezes os passos 2 e 3
5. Medir a posição da partícula $-t \leq x \leq t$

A distribuição de probabilidade da partícula será do tipo **binomial** e o seu desvio padrão é dado por:

$$\langle x^2 \rangle^{1/2} = \sqrt{t} \tag{45}$$

Caminho na reta - caminhante quântico

Partícula quântica com 2 graus de liberdade numa reta infinita.
Espaço de Hilbert: $H_2 \otimes H_\infty$.
$H_2 = \{|j\rangle, 0 \leq j \leq 1\}$. $H_\infty = \{|x\rangle, -\infty \leq x \leq \infty\}$ tal que x é inteiro.

1. Inicie sua partícula na origem: $x = 0$
2. Lance a moeda quântica $\mathbf{C}$ Responsável pela superposição.
   - $\mathbf{C}\,|0\rangle \otimes |x\rangle \longrightarrow \frac{(|0\rangle \otimes |x\rangle + |1\rangle \otimes |x\rangle)}{\sqrt{2}}$
   - $\mathbf{C}\,|1\rangle \otimes |x\rangle \longrightarrow \frac{(|0\rangle \otimes |x\rangle - |1\rangle \otimes |x\rangle)}{\sqrt{2}}$
3. Mova a partícula uma posição para esquerda e para direita de acordo com o estado da partícula
   - $\mathbf{S}\,|0\rangle \otimes |x\rangle \longrightarrow |0\rangle \otimes |x+1\rangle$
   - $\mathbf{S}\,|1\rangle \otimes |x\rangle \longrightarrow |1\rangle \otimes |x-1\rangle$
4. Repetir t vezes os passos 2 e 3
5. Medir a posição da partícula $-t \leq x \leq t$

A distribuição de probabilidade da partícula é bem curiosa e o seu desvio padrão é dado por:

$$\langle x^2 \rangle^{1/2} \propto t \tag{46}$$

Caminho na reta - Clássico versus Quântico

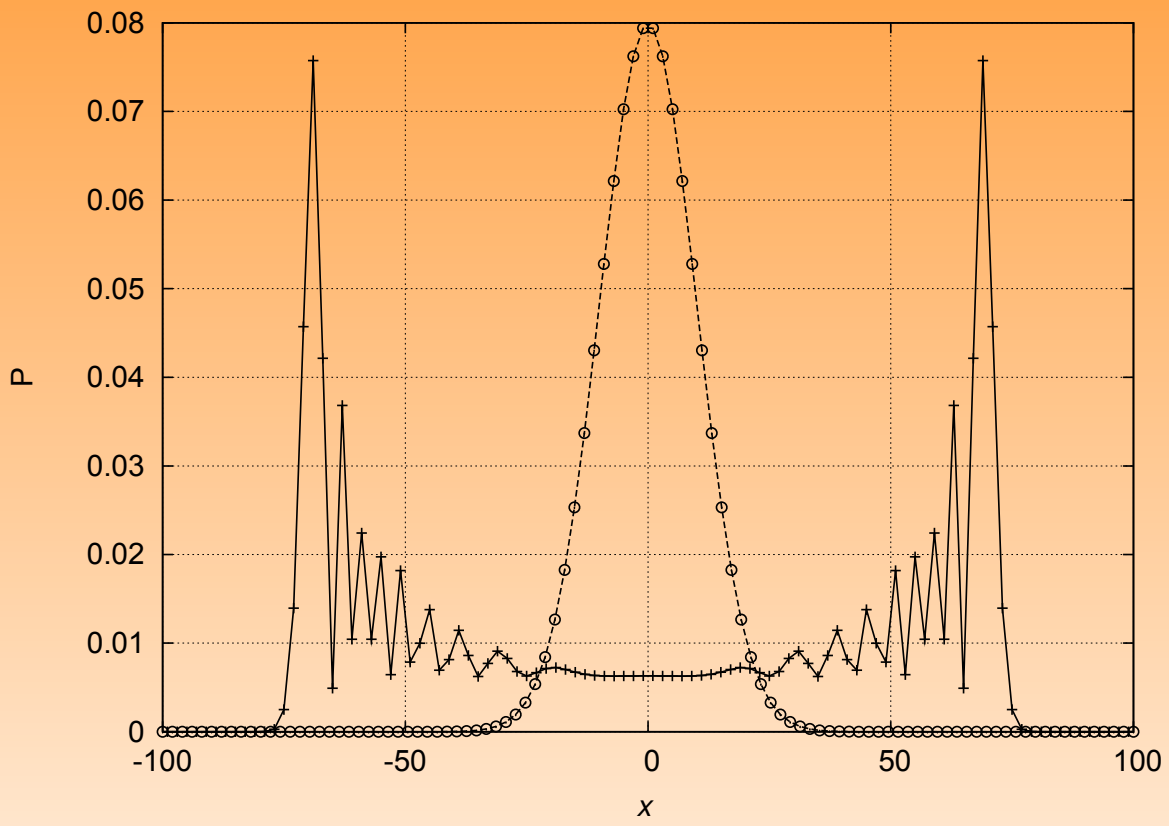


Figura: Distribuição de probabilidade do caso 1-D moeda de Hadamard e estado inicial $|\psi(0)\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \otimes |0\rangle$, versus o caso clássico após 100 passos.

O caminhante quântico propaga-se quadraticamente mais rápido!!.

Introdução à Computação Quântica

Figura: http://www.cs.umbc.edu/ lomonaco/qcomp/Qcomp.html

MUITO OBRIGADA - amanda@lncc.br

Agradecimentos

- Todo este material foi baseado principalmente no livro [1] que é a referência do curso.
- Agradeço aos autores R. Portugal, C. Lavor, L. M. Carvalho e N. Maculan pelo pioneirismo deste material tão didático em língua portuguesa e por todas as importantes contribuições na área.
- Agradeço imensamente aos colaboradores F.L. Marquezino, D.N. Gonçalves e E.B. Guedes pelas inúmeras colaborações na elaboração deste material.
- Agradeço à organização do curso de verão 2008 por todo apoio dispensado.


R. Portugal., C. Lavor, L. M. Carvalho, and N. Maculan.
Uma Introdução à Computação Quântica.
SBMAC,BR, 2004.


E. Knill, R. Laflamme, and W. Zurek.
Threshold accuracy for quantum computation.
1996.
lanl-arXive quant-ph/9610011.


D. Aharonov and M. Ben-Or.
Fault-tolerant quantum computation with constant error rate.
In *Proc. 29th. Ann. ACM Symp. on Theory of Computing*, 1997.
Longer version quant-ph/9906129.


A. Steane.
Multiple-particle interference and quantum error correction.
Proc. R. Soc. London, Ser. A, 452:2551, 1996.


R. J. Hughes, D. M. Alde, P. Dyer, G. G. Luther, G. L. Morgan, and M. Schauer.
Quantum cryptography.

Contemporary Physics, 36(3):149–163, 1995.


N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden.
Quantum cryptography.
Rev. Mod. Phys., 74(1):145–195, 2002.


P. Shor.
Polinomial-time algorithms for prime factorization and discrete logarithns on a quantum computer.
In *SIAM J. Comp*, volume 26(5), pages 1484–1509, 1997.


A.F. de Lima. and B.L.Júnior.
Computação Quântica noções básicas utilizando a linguagem de circuitos quânticos.
EDUFCG,BR, 2007.


L. K. Grover.
A fast quantum algorithm for database search.
In Proc. 28th Annual ACM Symposium on the Theory of Computing, pages 212–219, 1996.
lanl-arXive quant-ph/9605043.


L. K. Grover.
Quantum mechanics helps in searching for a needle in a haystack.
Phys. Rev. Lett., 79:325–328, 1997.
lanl-arXive quant-ph/9706033.


C.H.Papadimitriou.
Markov chains, Gibbs fields, Monte Carlo simulation, and queues.
Springer-Verlag, New York, 1999.


M. Dyer, A. Frieze, and R. Kannan.
A random polynomial-time algorithm for approximating the volume of convex bodies.
J. ACM, 38(1):1–17, January 1991.


M. Jerrum, A. Sinclair, and E. Vigoda.
A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries.
In *Proc. 33th STOC*, pages 712–721, New York, NY, 2001. ACM.


U. Schöning.
A probabilistic algorithm for k-sat and constraint satisfaction problems.

In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 410–414. IEEE, 1999.


R. Motwani and P. Raghavan.
Randomized Algorithms.
Cambridge University Press, UK, 1995.


Y. Aharonov, L. Davidovich, and N. Zagury.
Quantum random walks.
Phys. Rev. A, 48(2):1687–1690, 1993.


E. Farhi and S. Gutmann.
Quantum computation and decision trees.
Phys. Rev. A, 58:915–928, 1998.


D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani.
Quantum walks on graphs.
In *Proc. 33th STOC*, pages 50–59, New York, NY, 2001. ACM.