

Sumário

Introdução	1
Breve Histórico da Computação Quântica	1
Motivação para a Computação Quântica	2
Sobre o TCC	2
1 Álgebra Linear para Computação Quântica	5
1.1 Espaço Vetorial, Base e Dimensão	6
1.1.1 Espaço Vetorial	6
1.1.2 Base e Dimensão	7
1.1.3 Matriz de Mudança de Base	9
1.2 Produto Interno, Norma e Produto Exterior	10
1.2.1 Produto Interno	10
1.2.2 Norma	12
1.2.3 Ortogonalidade	12
1.2.4 Base Ortonormal	13
1.2.5 Desigualdade de Cauchy-Schwarz	14
1.2.6 Matriz de Mudança de Base entre Bases Ortonormais	15
1.2.7 Produto Exterior	16
1.3 Transformações Lineares	17
1.3.1 Transformação Linear e Operador Linear	17
1.3.2 Funcional Linear	18
1.3.3 Projeção e Relação de Completude	18
1.3.4 Definição de uma Transformação Linear nos Elementos da Base	19
1.3.5 Matriz de uma Transformação Linear	20

1.3.6	Matriz da Composição de Transformações Lineares	22
1.3.7	Mudança de Base	22
1.4	Autovalores, Autovetores e Decomposição Espectral	23
1.4.1	Autovalores e Autovetores	23
1.4.2	Cálculo de Autovalores	24
1.4.3	Cálculo de Autovetores	24
1.4.4	Diagonalização de Operadores	26
1.5	Tipos Especiais de Operadores	29
1.5.1	Operador Adjunto	29
1.5.2	Operadores Normais	30
1.5.3	Operadores Hermitianos ou Autoadjuntos	30
1.5.4	Operadores Unitários	31
1.5.5	Operadores Positivos	32
1.5.6	Operadores de Projeção	32
1.5.7	Resumo	34
1.6	Produto Tensorial	35
1.6.1	Espaço Vetorial do Produto Tensorial	35
1.6.2	Comparação do Produto Tensorial com o Produto Cartesiano	36
1.6.3	Produto Interno	37
1.6.4	Operadores	39
1.6.5	Produto de Kronecker	40
1.6.6	Produto Tensorial de Vários Espaços Vetoriais	41
1.6.7	Notação	41
2	Introdução à Mecânica Quântica	45
2.1	Postulados da Mecânica Quântica	45
2.1.1	Descrição de um Sistema Físico	45
2.1.2	Evolução Temporal de um Sistema Físico	46
2.1.3	Medidas em Sistemas Físicos	49
2.1.4	Valor esperado de um Observável	51
2.1.5	Sistemas Compostos	52
2.2	Matrizes de Pauli	54
2.2.1	Definição	55
2.2.2	Propriedades	55
2.2.3	Autovalores, Autovetores e Diagonalização	56
2.3	Estados de Bell	56
3	Computação Quântica	59
3.1	Computação Quântica de Circuitos	60
3.2	O Qubit	61
3.2.1	Descrição Matemática do Qubit	61

3.2.2	Fase Relativa e Fase Global	61
3.2.3	Representação de um Qubit na Esfera de Bloch . . .	63
3.2.4	Justificativas para a seção anterior	65
3.3	Notação de Circuitos	66
3.4	Portas Lógicas Quânticas	68
3.4.1	Portas Lógicas de 1 Qubit	68
3.4.2	Portas Lógicas de 2 Qubits	71
3.4.3	Portas Lógicas de 3 Qubits	74
3.5	Identities de Circuitos	75
3.6	Universalidade das Portas Lógicas Quânticas	85
3.6.1	Universalidade de Portas Lógicas na Computação Clássica Reversível	85
3.6.2	Universalidade de Portas Lógicas na Computação Quântica	86
3.7	Teorema da Não-Clonagem	87
4	Panorama Atual da Computação Quântica	91
4.1	Expectativas de Mercado	91
4.2	Empresas e Desenvolvimentos Atuais	94
5	Computação Quântica com IBM Quantum Experience	97
5.1	IBM Quantum Experience	97
5.1.1	Computadores Disponíveis	97
5.1.2	Como Programar	98
5.1.3	Informativos e Guias de Usuário	101
5.2	Circuito Quantum Half Adder	102
5.2.1	Projeto do Circuito	102
5.2.2	Simulação do Circuito no IBM QX	104
5.2.3	Execução do Circuito no IBM QX	105
6	Protocolos e Algoritmos Quânticos	107
6.1	Codificação Superdensa	107
6.1.1	Visão geral	107
6.1.2	Circuito	108
6.1.3	Funcionamento Detalhado	108
6.2	Circuito de Teletransporte	110
6.2.1	Visão Geral	111
6.2.2	Circuito	111
6.2.3	Funcionamento Detalhado	112
6.3	Oráculos Quânticos	115
6.3.1	Oráculo XOR	116
6.3.2	Oráculo de Fase	116

6.3.3	Construção do Oráculo de Fase usando o Oráculo XOR	117
6.4	Algoritmo de Deutsch-Jozsa	117
6.4.1	Problema de Deutsch-Jozsa	117
6.4.2	Algoritmo de Deutsch-Jozsa	118
6.4.3	Algoritmo Clássico	121
6.4.4	Comparação de Desempenho	126
6.5	Algoritmo de Simon	126
6.5.1	Problema de Simon	127
6.5.2	Algoritmo de Simon	129
6.5.3	Algoritmo Clássico	139
6.5.4	Comparação de Desempenho	141
6.6	Algoritmo de Busca de Grover	141
6.6.1	Problema de Grover	141
6.6.2	Algoritmo de Grover	142
6.6.3	Algoritmo Clássico	151
6.6.4	Comparação de Desempenho	151
6.7	Algoritmo de Bernstein-Vazirani	152
6.7.1	Problema de Bernstein-Vazirani	152
6.7.2	Algoritmo de Bernstein-Vazirani (versão XOR)	153
6.7.3	Algoritmo de Bernstein-Vazirani (versão fase)	155
6.7.4	Algoritmo Clássico	157
6.7.5	Comparação de Desempenho	157
6.8	Transformada de Fourier Quântica - QFT	158
6.8.1	Transformada Discreta de Fourier - DFT	158
6.8.2	Definição e Exemplos	158
6.8.3	Propriedades da QFT	158
6.8.4	Circuito para QFT	158
6.8.5	Algumas aplicações	158
6.9	Algoritmo de Shor	158
Conclusão		159
	Considerações Finais	159
	Perspectivas	159
Bibliografia		160
A Elementos de Computação Clássica		165
A.1	Introdução	165
A.2	Níveis de Abstração	166
A.3	Nível Lógico	167
A.3.1	Álgebra Booleana	167
A.3.2	Portas Lógicas	168

A.3.3	Teoremas da Álgebra Booleana	170
A.3.4	Universalidade das Portas Lógicas Clássicas	172
A.3.5	Somadores e Unidade Lógica/Aritmética	175

Introdução

Breve Histórico da Computação Quântica

No início do século XX, a Física Clássica enfrentava dificuldades em descrever alguns fenômenos observados à época, como, por exemplo, o espectro de radiação de corpo negro e o efeito fotoelétrico. Essa crise culminou na criação da Mecânica Quântica, que se consolidou por volta da década de 1920, e tem sido aplicada com sucesso em diversos fenômenos. ([15], p.2).

O desenvolvimento tecnológico a partir da década de 1970 permitiu o controle de sistemas quânticos individuais permitindo-se aprisionar átomos individuais em armadilhas (*traps*), isolando-os do restante do ambiente, e medindo seus diversos aspectos com precisão notável. Nesse contexto, passou-se a considerar a possibilidade de se usar sistemas quânticos para realizar processamento e transmissão de informação, fazendo uso de fundamentos da Mecânica Quântica, como superposição e emaranhamento. Esses sistemas guardam analogia com os bits clássicos, e são chamados de qubits. ([15] p.3-4).

O físico R. Feynman, na década de 1980, sugeriu o uso de computadores quânticos para simular sistemas quânticos. E em 1994, o matemático P. Shor propôs um algoritmo quântico capaz de resolver o problema de fatoração de números em fatores primos de forma mais eficiente que os algoritmos clássicos conhecidos. Há uma expectativa de que tal algoritmo possa ameaçar alguns protocolos de criptografia, como o RSA, usado largamente na atualidade. O cientista da computação Lov Grover também elaborou um algoritmo quântico de busca em uma base de dados não estruturada, que possui ganho quadrático de desempenho em comparação ao melhor algoritmo clássico conhecido. ([3], p.3).

Atualmente, a Computação Quântica e a Informação Quântica estão

se consolidando como áreas de pesquisa com desenvolvimento acelerado nas últimas décadas. Empresas de tecnologia como IBM, Google, Intel e Microsoft têm projetos e pesquisas nessa área, e diversas Startups têm surgido nesse contexto.

Motivação para a Computação Quântica

A *lei de Moore*, formulada em 1965, previu que a capacidade computacional dos sistemas digitais dobraria a cada dois anos. Surpreendentemente, os avanços computacionais se mantiveram aproximadamente nesse ritmo até a atualidade. No entanto, à medida que a escala dos transistores se reduz, aproxima-se de limites físicos fundamentais, e aparentemente insuperáveis. Os efeitos quânticos, em alguns aspectos indesejáveis (tal como o tunelamento), passam a interferir intensamente no funcionamento ideal do transistor. Para que o ritmo ditado pela lei de Moore continue, tem-se apostado, entre outras abordagens, na Computação Quântica. ([15], p.4 e [3], p.2).

Outro ponto refere-se ao consumo de energia. O princípio de Landauer afirma que para cada bit de informação apagado, dissipa-se no ambiente a energia correspondente a pelo menos $kT \ln 2$, em quem k é a constante de Boltzmann e T é a temperatura do sistema dada em Kelvin. Pelas características da Mecânica Quântica, as portas lógicas quânticas precisam ser reversíveis. Isso corresponderia, em princípio, à desnecessidade de se apagar informação e à possibilidade de se ter um computador que não dissipe energia no processamento. ([3], p. 3).

Um terceiro ponto motivador das pesquisas em Computação Quântica é que a investigação nessa área pode elucidar aspectos da Mecânica Quântica, frequentemente contra intuitivos em relação à Física Clássica, ou mesmo, apontar fenômenos ainda não explorados. ([15], p. 3).

Por fim, a Computação Quântica apresenta desafios em diversas áreas, como manipulação de sistemas quânticos, novas técnicas experimentais, desenvolvimento de algoritmos, teoria da informação, entre outros, o que a torna bastante atrativa e multidisciplinar.

Sobre o TCC

O presente trabalho tem como objetivo servir como um material introdutório e multidisciplinar sobre Computação Quântica. Visa-se com isso facilitar o primeiro contato com esse assunto por parte de estudantes de Engenharias, Ciências da Computação, Física, Matemática e áreas correlatas.

O conteúdo disposto neste trabalho começou a ser apresentado em sessões semanais com o orientador no primeiro semestre de 2017. Conforme mais pessoas juntaram-se às reuniões semanais, criou-se o Grupo de Computação Quântica da UFSC (GCQ-UFSC), e as reuniões tiveram objetivo de trazer um material introdutório aos estudantes de graduação do grupo e discutir atualidades da área de maneira acessível. As reuniões do GCQ-UFSC contribuíram grandemente para dar forma ao presente texto.

Os capítulos 1 e 2 contêm elementos de Álgebra Linear e Mecânica Quântica, pré-requisitos para o estudo da Computação Quântica, com ênfase nos pontos necessários para o restante do texto. Os capítulos 3 e 6 trazem uma introdução à Computação Quântica com uma abordagem voltada a algoritmos quânticos. O capítulo 4 contextualiza a Computação Quântica no cenário atual, discutindo expectativas de mercado e algumas das principais empresas envolvidas. Por fim, o capítulo 5 apresenta a plataforma IBM Quantum Experience, em que é possível escrever um algoritmo quântico e submeter para execução em um protótipo de computador quântico de 5 qubits, disponível em nuvem.

Capítulo 1

Álgebra Linear para Computação Quântica

Neste capítulo, um resumo de Álgebra Linear voltado para Computação Quântica é apresentado. A teoria é apresentada usando-se a *notação de Dirac*, ou, *notação de braket*, uma notação utilizada largamente em Mecânica Quântica e que será necessária para o restante do trabalho. Essa notação é conveniente para se fazer contas e faz com que as diversas operações possíveis se encaixem naturalmente.

Na Álgebra Linear abstrata, pode-se considerar espaços vetoriais sobre diversos corpos (ou escalares), que são estruturas algébricas com propriedades semelhantes aos números reais e complexos. Os espaços vetoriais podem ter dimensão finita ou infinita.

Na Computação Quântica o interesse é voltado a espaços vetoriais de dimensão finita sobre o corpo dos números complexos. Isso permite a identificação do espaço com as n -uplas de números complexos, o que simplifica grandemente a teoria. Os resultados resumidos neste capítulo estão situados nesse contexto.

A principal referência para esse capítulo é [15]. Outra referência muito útil é [10], que possui um capítulo voltado a espaços vetoriais complexos. Livros texto clássicos de Álgebra Linear, como [20] também são úteis. Embora tenham ênfase em espaços vetoriais reais, a maioria das definições, resultados e demonstrações se transporta integralmente para os espaços vetoriais complexos.

Será considerado um pré-requisito a este texto um curso de Álgebra Linear ao nível de graduação abordando-se os seguintes itens: espaços

vetoriais, base e dimensão, transformações lineares, autovalores e autovetores. Por ter um caráter de resumo, os resultados apresentados, via de regra, não são acompanhados de suas demonstrações, as quais podem ser encontradas nos livros mencionados no parágrafo anterior.

1.1 Espaço Vetorial, Base e Dimensão

1.1.1 Espaço Vetorial

O conjunto das n -uplas (z_0, \dots, z_{n-1}) de números complexos com a soma e o produto por escalar definidos entrada a entrada é um *Espaço Vetorial Complexo* e é denotado por \mathbb{C}^n . É conveniente representar esses elementos por vetores coluna. Tem-se então:

$$\begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_{n-1} \end{bmatrix} = \begin{bmatrix} z_0 + w_0 \\ z_1 + w_1 \\ \vdots \\ z_{n-1} + w_{n-1} \end{bmatrix} \quad \text{e} \quad z \cdot \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} z \cdot z_0 \\ z \cdot z_1 \\ \vdots \\ z \cdot z_{n-1} \end{bmatrix}.$$

Em Mecânica Quântica, os vetores de \mathbb{C}^n costumam ser usados na *notação de Dirac*, ou *notação de bracket*:

$$|\psi\rangle = (z_0, z_1, \dots, z_{n-1}) = \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix}.$$

Um vetor $|\psi\rangle$ é chamado *ket* (em contraponto com $\langle\psi|$, que será definido posteriormente, e será chamado *bra*).

Os vetores se comportam de maneira semelhante aos números no que diz respeito à soma e subtração. Em particular, a soma comuta $|\phi\rangle + |\psi\rangle = |\psi\rangle + |\phi\rangle$, há um vetor nulo, denotado por 0 ou $|\emptyset\rangle = (0, \dots, 0)$ de tal forma que $|\psi\rangle + 0 = |\psi\rangle$ e, ainda, vale $-|\psi\rangle = (-z_0, \dots, -z_{n-1}) = -1 \cdot |\psi\rangle$. O produto por escalar também se comporta de maneira semelhante ao produto numérico, e valem as propriedades distributivas $z(|\phi\rangle + |\psi\rangle) = z|\phi\rangle + z|\psi\rangle$ e $(z + w)|\psi\rangle = z|\psi\rangle + w|\psi\rangle$, por exemplo.

Exemplo 1.1 (Espaço de estados de 1 qubit). O conjunto

$$\mathbb{C}^2 = \left\{ |\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix} : a, b \in \mathbb{C} \right\}$$

é um espaço vetorial com soma e produto por escalar dados por

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} = \begin{bmatrix} a_1 + a_2 \\ b_1 + b_2 \end{bmatrix}$$

$$z \cdot \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} za \\ zb \end{bmatrix}.$$

Esse espaço vetorial será largamente utilizado nos capítulos seguintes e descreve o espaço de estados de 1 *qubit*, o análogo do bit clássico.

1.1.2 Base e Dimensão

Uma base para o espaço vetorial \mathbb{C}^n é um conjunto de vetores *linearmente independentes* (LI) e que *geram o espaço*. Demonstra-se que todas as bases de um espaço vetorial têm o mesmo número de elementos, e define-se a *dimensão* do espaço vetorial pelo número de elementos de uma base.

O espaço vetorial \mathbb{C}^n tem dimensão n , isto é, todas as suas bases têm n vetores. Uma base muito útil é a chamada *base computacional*, ou *base canônica*¹:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad \dots, \quad |n-1\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Na base computacional, um vetor $|\psi\rangle = (z_0, z_1, \dots, z_{n-1})$ é escrito como

$$|\psi\rangle = z_0 |0\rangle + z_1 |1\rangle + \dots + z_{n-1} |n-1\rangle.$$

Numa base qualquer $\beta = \{|b_0\rangle, \dots, |b_{n-1}\rangle\}$, qualquer vetor ψ pode ser escrito como combinação linear dos vetores dessa base. Os coeficientes da combinação linear, colocados em um vetor coluna, representam o vetor ψ escrito na base β , conforme o esquema abaixo:

$$|\psi\rangle = a_0 |b_0\rangle + \dots + a_{n-1} |b_{n-1}\rangle \Leftrightarrow [|\psi\rangle]_{\beta} = \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \end{bmatrix}_{\beta}.$$

O subscrito β pode ser omitido se não houver risco de confusão. Normalmente omite-se esse subscrito quando se trata da base computacional.

Exemplo 1.2 (Bases para 1 qubit). Há especial interesse no espaço \mathbb{C}^2 . Este espaço modela um *qubit* — o análogo quântico do bit — a ser discutido em mais detalhes posteriormente. O espaço \mathbb{C}^2 tem dimensão 2 e admite,

¹O adjetivo “canônico”, na Matemática, tem um sentido de “padrão”, como na expressão “configuração padrão”.

entre outras, as seguintes bases:

$$\begin{aligned}\mathcal{I} = \mathcal{Z} &= \{ |0\rangle, |1\rangle \} \\ \mathcal{X} &= \left\{ |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \right\} \\ \mathcal{Y} &= \left\{ |+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \right\} .\end{aligned}$$

Essa notação para as bases será justificada *a posteriori*.

Exemplo 1.3 (\mathcal{X} é base para 1 qubit). Para mostrar que

$$\mathcal{X} = \left\{ |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \right\}$$

é base do espaço de estados de 1 qubit, deve-se mostrar que os vetores são LI (Linearmente Independentes) e geram o espaço \mathbb{C}^2 .

\mathcal{X} é LI: Considere a combinação linear nula:

$$\begin{aligned}a_0 |+\rangle + a_1 |-\rangle &= 0 \\ a_0 \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + a_1 \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) &= 0 \\ \frac{a_0 + a_1}{\sqrt{2}} |0\rangle + \frac{a_0 - a_1}{\sqrt{2}} |1\rangle &= 0\end{aligned}$$

Tem-se que

$$\begin{cases} \frac{a_0 + a_1}{\sqrt{2}} = 0 \\ \frac{a_0 - a_1}{\sqrt{2}} = 0 \end{cases} \implies \begin{cases} a_0 + a_1 = 0 \\ a_0 - a_1 = 0 \end{cases} \implies \begin{cases} a_0 = 0 \\ a_1 = 0 \end{cases} .$$

Portanto os coeficientes da combinação linear nula devem ser todos nulos, e isso significa que os vetores $|+\rangle$ e $|-\rangle$ são LI.

\mathcal{X} gera o espaço: Seja $|\psi\rangle = z_0 |0\rangle + z_1 |1\rangle$ um vetor qualquer de \mathbb{C}^2 . Tenta-se escrever $|\psi\rangle$ como combinação linear de $|+\rangle$ e $|-\rangle$. Se for possível, esses vetores geram o espaço.

$$\begin{aligned}z_0 |0\rangle + z_1 |1\rangle &= a_0 |+\rangle + a_1 |-\rangle \\ &= a_0 \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) + a_1 \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ &= \frac{a_0 + a_1}{\sqrt{2}} |0\rangle + \frac{a_0 - a_1}{\sqrt{2}} |1\rangle\end{aligned}$$

Assim,

$$\begin{cases} \frac{a_0 + a_1}{\sqrt{2}} = z_0 \\ \frac{a_0 - a_1}{\sqrt{2}} = z_1 \end{cases} \implies \begin{cases} a_0 = \frac{z_0 + z_1}{\sqrt{2}} \\ a_1 = \frac{z_0 - z_1}{\sqrt{2}} \end{cases}$$

Dessa forma, \mathcal{X} gera o espaço \mathbb{C}^2 e é base desse espaço.

Comentário: Quando se sabe previamente que a dimensão do espaço é n e se a lista de vetores candidatos a base tem n elementos, então as condições de ser LI e gerar o espaço são equivalentes. Em consequência, basta verificar uma das condições para mostrar que os vetores formam uma base. Por exemplo, se sabemos que a dimensão de \mathbb{C}^2 é $\dim \mathbb{C}^2 = 2$, e temos que \mathcal{X} tem dois elementos, então bastaria verificar uma das duas condições: \mathcal{X} é LI ou \mathcal{X} gera o espaço.

1.1.3 Matriz de Mudança de Base

Por vezes é conveniente expressar um vetor em outra base que não a base computacional. Essa mudança de base pode trazer novo *insight* em algumas situações, bem como tornar os cálculos mais simples e factíveis.

A matriz de mudança de base de $\beta_{\text{old}} = \{|u_0\rangle, \dots, |u_{n-1}\rangle\}$ para $\beta_{\text{new}} = \{|v_0\rangle, \dots, |v_{n-1}\rangle\}$ é dada por

$$[I]_{\beta_{\text{new}}}^{\beta_{\text{old}}} = \begin{bmatrix} | & & \\ [|v_0\rangle]_{\beta_{\text{old}}} & \cdots & [|v_{n-1}\rangle]_{\beta_{\text{old}}} \\ | & & \end{bmatrix}.$$

Isto é, para montar a matriz de mudança de base, os vetores da base nova devem ser escritos como combinação linear dos vetores da base antiga, obtendo vetores coluna. Esses vetores serão as colunas da matriz de mudança de base. Dessa forma, tem-se

$$[v]_{\beta_{\text{new}}} = [I]_{\beta_{\text{new}}}^{\beta_{\text{old}}} [v]_{\beta_{\text{old}}}.$$

A matriz de mudança de base admite matriz inversa, que corresponde à mudança de base da base nova de volta para a base antiga:

$$[I]_{\beta_{\text{new}}}^{\beta_{\text{old}}}{}^{-1} = [I]_{\beta_{\text{old}}}^{\beta_{\text{new}}}.$$

Observação 1.4. O símbolo I na matriz de mudança de base $[I]_{\beta_{\text{new}}}^{\beta_{\text{old}}}$ refere-se ao operador identidade. A matriz de mudança de base corresponde à matriz do operador identidade nas bases β_{old} e β_{new} . As matrizes referente a operadores lineares serão vistas na seção 1.3.5.

Exemplo 1.5. Considere as bases \mathcal{I} e \mathcal{X} apresentadas no exemplo 1.2. A matriz de mudança de base da base computacional \mathcal{I} para a base \mathcal{X} é obtida escrevendo os vetores da base nova (\mathcal{X}) como combinação linear dos

vetores na base antiga (\mathcal{I}):

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}_{\mathcal{I}} \\ |-\rangle &= \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}_{\mathcal{I}} \end{aligned}$$

Colocam-se as os vetores coluna na ordem em que aparecem na lista:

$$[I]_{\mathcal{X}}^{\mathcal{I}} = \begin{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}_{\mathcal{I}} & \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}_{\mathcal{I}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Essa matriz é conhecida em Computação Quântica como *matriz de Hadamard*, e costuma ser denotada por H . Portanto

$$H = [I]_{\mathcal{X}}^{\mathcal{I}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Também vale que a matriz de mudança de base de \mathcal{X} para \mathcal{I} é H , visto que

$$HH = I \implies H^{-1} = H \implies [I]_{\mathcal{I}}^{\mathcal{X}} = ([I]_{\mathcal{X}}^{\mathcal{I}})^{-1} = H^{-1} = H.$$

Então:

$$\begin{aligned} H|0\rangle &= |+\rangle & H|+\rangle &= |0\rangle \\ H|1\rangle &= |-\rangle & H|-\rangle &= |1\rangle \end{aligned}.$$

1.2 Produto Interno, Norma e Produto Exterior

1.2.1 Produto Interno

O espaço vetorial \mathbb{C}^n admite o seguinte *produto interno*:

$$(|\phi\rangle, |\psi\rangle) = \langle\phi|\psi\rangle = [w_0^* \ w_1^* \ \cdots \ w_{n-1}^*] \cdot \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{bmatrix} = \sum_{k=0}^{n-1} w_k^* z_k, \quad ,$$

para $|\phi\rangle = \begin{bmatrix} w_0 \\ \vdots \\ w_{n-1} \end{bmatrix}$ e $|\psi\rangle = \begin{bmatrix} z_0 \\ \vdots \\ z_{n-1} \end{bmatrix}$. Pode-se considerar o símbolo $\langle\phi|$ de maneira independente, definindo-o como:

$$\langle\phi| = |\phi\rangle^\dagger = \begin{bmatrix} w_0 \\ \vdots \\ w_{n-1} \end{bmatrix}^\dagger = [w_0^* \ w_1^* \ \cdots \ w_{n-1}^*], \quad ,$$

em que o símbolo \dagger denota transposição e conjugação do vetor. Essa notação também será justificada posteriormente.

A operação definida acima satisfaz as propriedades que definem um produto interno de maneira geral:

(PI1) Linearidade no segundo argumento:

$$(|\phi\rangle, z_1 |\psi_1\rangle + z_2 |\psi_2\rangle) = z_1 (|\phi\rangle, |\psi_1\rangle) + z_2 (|\phi\rangle, |\psi_2\rangle)$$

(PI2) Antilinearidade no primeiro argumento:

$$(z_1 |\phi_1\rangle + z_2 |\phi_2\rangle, |\psi\rangle) = z_1^* (|\phi_1\rangle, |\psi\rangle) + z_2^* (|\phi_2\rangle, |\psi\rangle)$$

(PI3) Simetria hermitiana:

$$(|\phi\rangle, |\psi\rangle)^* = (|\psi\rangle, |\phi\rangle)$$

(PI4) Positividade:

$$(|\phi\rangle, |\phi\rangle) \geq 0 \quad \text{e} \quad (|\phi\rangle, |\phi\rangle) = 0 \Leftrightarrow |\phi\rangle = 0$$

A propriedade (PI2) decorre de (PI1) e de (PI3), mas foi incluída na lista por completeza. Por causa das propriedades (PI1) e (PI2), o produto interno é dito ser *sesquilinear*².

O espaço vetorial \mathbb{C}^n é, pois, dito ser um *espaço vetorial com produto interno*, ou ainda, um *espaço de Hilbert*³.

Exemplo 1.6. O produto interno dos vetores

$$|\phi\rangle = \frac{i}{2} |0\rangle + \frac{\sqrt{3}}{2} |1\rangle = \begin{bmatrix} \frac{i}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad \text{e} \quad |\psi\rangle = \frac{1}{\sqrt{2}} |0\rangle + \frac{-i}{\sqrt{2}} |1\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{bmatrix}$$

é dado por:

$$\begin{aligned} \langle\phi|\psi\rangle &= \left(\frac{-i}{2} \langle 0| + \frac{\sqrt{3}}{2} \langle 1| \right) \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{-i}{\sqrt{2}} |1\rangle \right) \\ &= \frac{-i}{2\sqrt{2}} \langle 0|0\rangle + \frac{-i^2}{2\sqrt{2}} \langle 0|1\rangle + \frac{\sqrt{3}}{2\sqrt{2}} \langle 1|0\rangle + \frac{-i\sqrt{3}}{2\sqrt{2}} \langle 0|0\rangle \\ &= \frac{-i}{2\sqrt{2}} + 0 + 0 + \frac{-i\sqrt{3}}{2\sqrt{2}} = -i \frac{1 + \sqrt{3}}{2\sqrt{2}}. \end{aligned}$$

²O prefixo *sesqui* significa “um e meio”.

³Um espaço de Hilbert é definido como sendo um espaço vetorial com produto interno e com uma propriedade adicional chamada *completez*. Essa propriedade é automática para espaços de dimensão finita.

Esse produto interno também pode ser calculado de maneira matricial:

$$\begin{aligned}
 \langle \phi | \psi \rangle &= \begin{bmatrix} \frac{i}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}^\dagger \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{i}{2}^* & \frac{\sqrt{3}}{2}^* \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{-i}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{-i}{\sqrt{2}} \end{bmatrix} \\
 &= \frac{-i}{2\sqrt{2}} + \frac{-i\sqrt{3}}{2\sqrt{2}}
 \end{aligned}$$

1.2.2 Norma

A *norma*, ou *tamanho*, de um vetor é definida por

$$\| |\psi\rangle \| = \sqrt{\langle \psi | \psi \rangle} \geq 0 \quad ,$$

operação bem definida pois $\langle \psi | \psi \rangle$ é real e não-negativo. Por consequência da propriedade (P4), tem-se $\| |\psi\rangle \| = 0 \Leftrightarrow |\psi\rangle = 0$. Um vetor *normalizado* é um vetor de tamanho unitário, e a operação de *normalização* consiste em multiplicar o vetor $|\psi\rangle$ por $\frac{1}{\| |\psi\rangle \|}$ para que o vetor resultante $\frac{|\psi\rangle}{\| |\psi\rangle \|}$ tenha norma 1.

Exemplo 1.7. Os vetores $|0\rangle$, $|1\rangle$, $|+\rangle$ e $|-\rangle$ têm norma 1.

Exemplo 1.8. A norma do vetor $|\psi\rangle = \frac{i}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$ é

$$\| |\psi\rangle \| = \sqrt{\left| \frac{i}{2} \right|^2 + \left| \frac{\sqrt{3}}{2} \right|^2} = \sqrt{\frac{1}{4} + \frac{3}{4}} = 1 \quad .$$

1.2.3 Ortogonalidade

De forma análoga com o que se passa com vetores no \mathbb{R}^3 , dois vetores $|\phi\rangle$ e $|\psi\rangle$ são ditos *ortogonais* se o produto interno entre eles é nulo. Em símbolos, $|\phi\rangle \perp |\psi\rangle \Leftrightarrow \langle \phi | \psi \rangle = 0$.

Exemplo 1.9. Os vetores $|0\rangle$ e $|1\rangle$ são ortogonais:

$$\langle 0 | 1 \rangle = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 \quad .$$

Exemplo 1.10. Os vetores $|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ e $|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ são ortogonais:

$$\begin{aligned}\langle + | - \rangle &= \left(\frac{1}{\sqrt{2}} \langle 0 | + \frac{1}{\sqrt{2}} \langle 1 | \right) \left(\frac{1}{\sqrt{2}} | 0 \rangle - \frac{1}{\sqrt{2}} | 1 \rangle \right) \\ &= \frac{1}{2} \langle 0 | 0 \rangle - \frac{1}{2} \langle 0 | 1 \rangle + \frac{1}{2} \langle 1 | 0 \rangle - \frac{1}{2} \langle 1 | 1 \rangle \\ &= \frac{1}{2} - 0 + 0 - \frac{1}{2} = 0 .\end{aligned}$$

1.2.4 Base Ortonormal

Uma base em que todos os vetores são ortogonais dois a dois e têm tamanho 1 é dita ser *base ortonormal*. Se $\beta = \{|b_0\rangle, \dots, |b_{n-1}\rangle\}$ é uma tal base, vale a chamada *relação de ortogonalidade*

$$\langle b_k | b_l \rangle = \delta_{k,l} = \begin{cases} 1, & \text{se } k = l \\ 0, & \text{se } k \neq l \end{cases} ,$$

em que $\delta_{k,l}$ é conhecido como *delta de Kronecker*, e vale 1 se e somente se os seus dois índices são iguais; se forem diferentes, vale 0.

Um vetor $|\psi\rangle$ pode ser escrito como combinação linear dos vetores da base β por $|\psi\rangle = \sum_k a_k |b_k\rangle$. Os coeficientes a_k podem ser encontrados de maneira simples quando a base é ortonormal. Aplicando-se o produto interno em ambos os lados, tem-se

$$\langle b_l | \psi \rangle = \langle b_l | \left(\sum_k a_k |b_k\rangle \right) = \sum_k a_k \langle b_l | b_k \rangle = \sum_k a_k \delta_{l,k} = a_l .$$

Percebe-se que isso é análogo à decomposição de um vetor $\vec{v} \in \mathbb{R}^3$ nas suas componentes x , y e z na base canônica. Nesse caso, as componentes do vetor são dadas por $v_x = \hat{x} \cdot \vec{v}$, $v_y = \hat{y} \cdot \vec{v}$ e $v_z = \hat{z} \cdot \vec{v}$, em que \cdot denota o produto interno no \mathbb{R}^3 .

Portanto, os coeficientes do vetor $|\psi\rangle$ na base ortonormal são obtidos realizando-se projeções de $|\psi\rangle$ na direção dos vetores unitários $|b_l\rangle$ da base ortonormal. Assim:

$$|\psi\rangle = \sum_k \langle b_k | \psi \rangle |b_k\rangle \Leftrightarrow [|\psi\rangle]_\beta = \begin{bmatrix} \langle b_0 | \psi \rangle \\ \vdots \\ \langle b_{n-1} | \psi \rangle \end{bmatrix}_\beta .$$

Exemplo 1.11. A base $|0\rangle, |1\rangle$ é ortonormal, em consequência dos exemplos 1.7 e 1.9. As projeções de um vetor $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$ na base

computacional são dadas por

$$\begin{aligned} a_0 &= \langle 0|\psi\rangle \\ a_1 &= \langle 1|\psi\rangle \end{aligned}$$

e o vetor $|\psi\rangle$ pode ser escrito como

$$|\psi\rangle = \langle 0|\psi\rangle |0\rangle + \langle 1|\psi\rangle |1\rangle .$$

Exemplo 1.12. A base $|+\rangle, |-\rangle$ é ortonormal em consequência dos exemplos 1.7 e 1.10. Os coeficientes do vetor $|\psi\rangle = \frac{1}{2}(|0\rangle + i\sqrt{3}|1\rangle)$ na base \mathcal{X} são dados por:

$$\begin{aligned} a_0 &= \langle +|\psi\rangle \\ &= \frac{1}{\sqrt{2}} \frac{1}{2} (\langle 0| + \langle 1|) (|0\rangle + i\sqrt{3}|1\rangle) \\ &= \frac{1}{2\sqrt{2}}(1 + i\sqrt{3}) \\ a_1 &= \langle -|\psi\rangle \\ &= \frac{1}{\sqrt{2}} \frac{1}{2} (\langle 0| - \langle 1|) (|0\rangle + i\sqrt{3}|1\rangle) \\ &= \frac{1}{2\sqrt{2}}(1 - i\sqrt{3}) \end{aligned}$$

Portanto,

$$|\psi\rangle = \langle +|\psi\rangle |+\rangle + \langle -|\psi\rangle |-\rangle = \frac{1 + i\sqrt{3}}{2\sqrt{2}} |+\rangle + \frac{1 - i\sqrt{3}}{2\sqrt{2}} |-\rangle .$$

1.2.5 Desigualdade de Cauchy-Schwarz

Há uma desigualdade envolvendo normas de vetores que é válida de maneira geral, para quaisquer espaços vetoriais equipados com produto interno e norma. É conhecida por *desigualdade de Cauchy-Schwarz*.

Teorema 1.13 (Desigualdade de Cauchy-Schwarz). *Dados $|u\rangle, |v\rangle \in V$, com V um espaço vetorial qualquer munido de produto interno e norma, vale que*

$$|\langle u|v\rangle| \leq \| |u\rangle \| \| |v\rangle \| .$$

A igualdade ocorre se e somente se os vetores $|u\rangle$ e $|v\rangle$ forem múltiplos um do outro.

Demonstração. Considere o vetor $a|u\rangle - b|v\rangle$. O produto interno desse vetor consigo mesmo deve ser real e não-negativo, por propriedade do produto interno, portanto

$$\begin{aligned} 0 &\leq (a^* \langle u| - b^* \langle v|)(a|u\rangle - b|v\rangle) \\ &= |a|^2 \langle u|u\rangle - a^* b \langle u|v\rangle - ab^* \langle v|u\rangle + |b|^2 \langle v|v\rangle. \end{aligned}$$

Escolhendo $a = \langle v|v\rangle$ e $b = \langle v|u\rangle$, tem-se

$$\begin{aligned} 0 &\leq |a|^2 \langle u|u\rangle - a^* b \langle u|v\rangle - ab^* \langle v|u\rangle + |b|^2 \langle v|v\rangle \\ &= |\langle v|v\rangle|^2 \langle u|u\rangle - \langle v|v\rangle \langle v|u\rangle \langle u|v\rangle - \langle v|v\rangle \langle u|v\rangle \langle v|u\rangle + |\langle v|u\rangle|^2 \langle v|v\rangle \\ &= \langle v|v\rangle^2 \langle u|u\rangle - 2 \langle v|v\rangle \langle u|v\rangle \langle v|u\rangle + |\langle u|v\rangle|^2 \langle v|v\rangle \\ &= \langle v|v\rangle^2 \langle u|u\rangle - 2 \langle v|v\rangle |\langle u|v\rangle|^2 + |\langle u|v\rangle|^2 \langle v|v\rangle \end{aligned}$$

Logo, obtém-se

$$\begin{aligned} 0 &\leq \langle u|u\rangle \langle v|v\rangle - |\langle u|v\rangle|^2 \\ \implies |\langle u|v\rangle| &\leq \sqrt{\langle u|u\rangle \langle v|v\rangle} \\ \implies |\langle u|v\rangle| &\leq \|u\| \|v\|, \end{aligned}$$

e fica provada a desigualdade.

Ocorre igualdade se e somente se ocorrer igualdade em $0 \leq (a^* \langle u| - b^* \langle v|)(a|u\rangle - b|v\rangle)$, isto é, se e somente se o vetor $a|u\rangle - b|v\rangle$ for nulo⁴. Portanto deve-se ter um dos vetores $|u\rangle, |v\rangle$ múltiplo do outro. \square

No espaço vetorial \mathbb{R}^n essa desigualdade permite definir o conceito de ângulo entre dois vetores por meio da relação $\cos \theta = \frac{\overrightarrow{u} \cdot \overrightarrow{v}}{\|\overrightarrow{u}\| \|\overrightarrow{v}\|}$. No caso de interesse para o presente trabalho, o do espaço vetorial \mathbb{C}^n , essa interpretação não é possível pois o produto interno pode assumir um valor complexo.

1.2.6 Matriz de Mudança de Base entre Bases Ortonormais

Da mesma forma que em 1.1.3, é possível escrever uma matriz que faz a mudança de base entre duas bases ortonormais $\beta_{\text{old}} = \{|u_0\rangle, \dots, |u_{n-1}\rangle\}$ e $\beta_{\text{new}} = \{|v_0\rangle, \dots, |v_{n-1}\rangle\}$. As colunas da matriz de mudança de base são os vetores da base nova escritos como combinação linear dos vetores da base antiga. Como as bases são ortonormais, esses coeficientes podem

⁴Isso é devido à propriedade (PI4) do produto interno.

ser obtidos pela projeção na direção dos vetores da base, como descrito na seção 1.2.4.

A matriz de mudança de base, nesse caso, é dada por

$$\begin{aligned} [I]_{\beta_{\text{new}}}^{\beta_{\text{old}}} &= \begin{bmatrix} \left[\begin{array}{c} | \\ |v_0\rangle \end{array} \right]_{\beta_{\text{old}}} & \left[\begin{array}{c} | \\ |v_1\rangle \end{array} \right]_{\beta_{\text{old}}} & \cdots & \left[\begin{array}{c} | \\ |v_{n-1}\rangle \end{array} \right]_{\beta_{\text{old}}} \end{bmatrix} \\ &= \begin{bmatrix} \langle u_0|v_0\rangle & \langle u_0|v_1\rangle & \cdots & \langle u_0|v_{n-1}\rangle \\ \langle u_1|v_0\rangle & \langle u_1|v_1\rangle & \cdots & \langle u_1|v_{n-1}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle u_{n-1}|v_0\rangle & \langle u_{n-1}|v_1\rangle & \cdots & \langle u_{n-1}|v_{n-1}\rangle \end{bmatrix}. \end{aligned}$$

A matriz de mudança de base satisfaz $[I]_{\beta_{\text{new}}}^{\beta_{\text{old}}}{}^{-1} = [I]_{\beta_{\text{new}}}^{\beta_{\text{old}}}{}^{\dagger} = [I]_{\beta_{\text{old}}}^{\beta_{\text{new}}}$. Isso significa que a matriz de mudança de base é *unitária*, um assunto que será visto na seção 1.5.4.

Exemplo 1.14. Sabendo que as bases \mathcal{I} e \mathcal{X} são ortonormais, pode-se encontrar a matriz de mudança de base fazendo as seguintes contas.

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

Portanto, a matriz de mudança de base de \mathcal{I} para \mathcal{X} é dada por:

$$[I]_{\mathcal{X}}^{\mathcal{I}} = \begin{bmatrix} \langle 0|+\rangle & \langle 0|-\rangle \\ \langle 1|+\rangle & \langle 1|-\rangle \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = H.$$

Esse é o mesmo resultado obtido no exemplo 1.5.

1.2.7 Produto Exterior

É possível dar um significado ao símbolo $\langle\phi|$ como sendo um vetor linha. Isso permite, pois, considerar-se o produto matricial $|\psi\rangle \cdot \langle\phi| = |\psi\rangle\langle\phi|$ de um vetor linha $n \times 1$ por um vetor coluna $1 \times n$, resultando em uma matriz $n \times n$. Essa operação é chamada de *produto exterior*.

Exemplo 1.15. No espaço vetorial de 1 qubit, alguns exemplos de produto exterior são:

$$|0\rangle\langle 0| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$|0\rangle\langle 1| = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$|1\rangle\langle 0| = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

$$|1\rangle\langle 1| = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

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

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

1.3 Transformações Lineares

1.3.1 Transformação Linear e Operador Linear

Uma *transformação linear* é uma aplicação $T: \mathbb{C}^n \rightarrow \mathbb{C}^m$ que respeita a soma e a multiplicação por escalar, ou seja, tal que valem:

(TL1) Preservação da soma:

$$T(|\phi\rangle + |\psi\rangle) = T|\phi\rangle + T|\psi\rangle$$

(TL2) Preservação do produto por escalar:

$$T(z|\psi\rangle) = z \cdot T|\psi\rangle.$$

Um *operador linear* é uma transformação linear $A: \mathbb{C}^n \rightarrow \mathbb{C}^n$ ($m = n$).

Exemplo 1.16. A função $H: \mathbb{C}^2 \rightarrow \mathbb{C}^2$ dada por

$$H(a_0|0\rangle + a_1|1\rangle) = \frac{a_0 + a_1}{\sqrt{2}}|0\rangle + \frac{a_0 - a_1}{\sqrt{2}}$$

é uma transformação linear. De fato, as propriedades de transformação linear se verificam para H .

Preservação da soma: Sejam $|\phi\rangle = a_0 |0\rangle + a_1 |1\rangle$ e $|\psi\rangle = b_0 |0\rangle + b_1 |1\rangle$.

$$\begin{aligned}
 H(|\phi\rangle + |\psi\rangle) &= H(a_0 |0\rangle + a_1 |1\rangle + b_0 |0\rangle + b_1 |1\rangle) \\
 &= H((a_0 + b_0) |0\rangle + (a_1 + b_1) |1\rangle) \\
 &= \frac{(a_0 + b_0) + (a_1 + b_1)}{\sqrt{2}} |0\rangle + \frac{(a_0 + b_0) - (a_1 + b_1)}{\sqrt{2}} |1\rangle \\
 &= \frac{a_0 + a_1}{\sqrt{2}} |0\rangle + \frac{a_0 - a_1}{\sqrt{2}} |1\rangle + \frac{b_0 + b_1}{\sqrt{2}} |0\rangle + \frac{b_0 - b_1}{\sqrt{2}} |1\rangle \\
 &= H|\phi\rangle + H|\psi\rangle
 \end{aligned}$$

Preservação do produto por escalar: Sejam $z \in \mathbb{C}$ e $|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle$.

$$\begin{aligned}
 H(z|\psi\rangle) &= H(z(a_0 |0\rangle + a_1 |1\rangle)) \\
 &= H(za_0 |0\rangle + za_1 |1\rangle) \\
 &= \frac{za_0 + za_1}{\sqrt{2}} |0\rangle + \frac{za_0 - za_1}{\sqrt{2}} |1\rangle \\
 &= z \left(\frac{a_0 + a_1}{\sqrt{2}} |0\rangle + \frac{a_0 - a_1}{\sqrt{2}} |1\rangle \right) \\
 &= z \cdot H|\psi\rangle
 \end{aligned}$$

1.3.2 Funcional Linear

Um *funcional linear* é uma transformação linear $f: \mathbb{C}^n \rightarrow \mathbb{C}$ ($m = 1$). O bra $\langle\phi|$ pode ser pensado como um funcional linear que pode atuar em um vetor coluna $|\psi\rangle$ para resultar no número complexo $\langle\phi|\psi\rangle$. Pode-se verificar que todo funcional linear é da forma $\langle\phi| = \langle\phi|\cdot\rangle$ para algum $|\phi\rangle$.

Exemplo 1.17. O bra $\langle 0|$ é um funcional linear que leva $|\psi\rangle$ no coeficiente $\langle 0|\psi\rangle$ da projeção na direção $|0\rangle$. Igualmente, o bra $\langle 1|$ é um funcional linear que leva $|\psi\rangle$ no coeficiente $\langle 1|\psi\rangle$ da projeção na direção $|1\rangle$.

1.3.3 Projeção e Relação de Completude

Se $|u\rangle$ for unitário, o funcional linear $\langle u|$ leva um ket $|\psi\rangle$ em $\langle u|\psi\rangle$, que corresponde ao coeficiente da projeção de $|\psi\rangle$ na direção de $|u\rangle$.

O vetor $\langle u|\psi\rangle |u\rangle$ é a projeção de $|\psi\rangle$ na direção do vetor unitário $|u\rangle$. Movendo-se o número $\langle u|\psi\rangle$ para a direita, pode-se escrever essa projeção como $\text{proj}_{|u\rangle} |\psi\rangle = |u\rangle \langle u|\psi\rangle$. O operador $|u\rangle \langle u|$ é, então, chamado operador projeção na direção de $|u\rangle$.

Se $\beta = \{|b_0\rangle, \dots, |b_{n-1}\rangle\}$ é uma base ortonormal, pode-se escrever qualquer vetor $|\psi\rangle$ como soma das suas projeções ortogonais sobre as direções

definidas pelos vetores da base. Dessa forma, tem-se

$$|\psi\rangle = \sum_{k=0}^{n-1} \langle b_k | \psi \rangle |b_k\rangle = \sum_{k=0}^{n-1} |b_k\rangle \langle b_k | \psi \rangle \quad .$$

Segue que

$$\sum_{k=0}^{n-1} |b_k\rangle \langle b_k| = I \quad ,$$

expressão conhecida como *relação de completude*.

Exemplo 1.18. A projeção ortogonal da direção do vetor $|0\rangle$ é o operador $|0\rangle\langle 0|$, visto que sua ação em um ket $|\psi\rangle$ é dada por $|0\rangle\langle 0| |\psi\rangle = \langle 0 | \psi \rangle |0\rangle$ e a projeção ortogonal na direção de $|1\rangle$ é o operador de projeção $|1\rangle\langle 1|$, pois $|1\rangle\langle 1| |\psi\rangle = \langle 1 | \psi \rangle |1\rangle$.

A relação de completude no espaço vetorial dos estados de 1 qubit, \mathbb{C}^2 , é dada por

$$I = |0\rangle\langle 0| + |1\rangle\langle 1| \quad .$$

1.3.4 Definição de uma Transformação Linear nos Elementos da Base

Para definir uma transformação linear, basta que se forneça como ela atua nos elementos de uma base. Isto é, dada $\beta = \{|b_0\rangle, \dots, |b_{n-1}\rangle\}$ base de \mathbb{C}^n , pode-se obter $T|\psi\rangle$ conhecendo-se $T|b_k\rangle$ para todo k . De fato, como β base, pode-se escrever $|\psi\rangle$ como combinação linear

$$|\psi\rangle = \sum_k a_k |b_k\rangle \quad .$$

Aplicando-se a transformação T e usando a linearidade, obtém-se

$$T|\psi\rangle = \sum_k a_k T|b_k\rangle \quad ,$$

e dessa forma, $T|\psi\rangle$ pode ser obtido a partir dos $T|b_k\rangle$'s.

Exemplo 1.19. Considere o operador linear em 1 qubit $X: \mathbb{C}^2 \rightarrow \mathbb{C}^2$ definido nos vetores da base computacional por

$$\begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned}$$

O operador X está bem definido em todo $|\psi\rangle = a|0\rangle + b|1\rangle$ graças à sua linearidade:

$$X|\psi\rangle = X(a|0\rangle + b|1\rangle) = aX|0\rangle + bX|1\rangle = a|1\rangle + b|0\rangle \quad .$$

1.3.5 Matriz de uma Transformação Linear

Seja $T: U = \mathbb{C}^n \rightarrow V = \mathbb{C}^m$ uma transformação linear. Sejam $\beta_U = \{|u_0\rangle, \dots, |u_{n-1}\rangle\}$ base de U e $\beta_V = \{|v_0\rangle, \dots, |v_{m-1}\rangle\}$ base de V . Quando fixadas bases para os espaços vetoriais do domínio e do contradomínio de T , é possível representar a transformação T por uma matriz, de forma que a atuação de T sobre um vetor $|\psi\rangle$ é equivalente ao produto matriz-vetor coluna.

A matriz da transformação linear T nas bases β_U e β_V é dada por:

$$[T]_{\beta_V}^{\beta_U} = \begin{bmatrix} | & & | \\ [T(u_0)]_{\beta_V} & \cdots & [T(u_{n-1})]_{\beta_V} \\ | & & | \end{bmatrix} \in M(m, n)$$

Definida dessa forma, vale que:

$$[T|\psi\rangle]_{\beta_V} = [T]_{\beta_V}^{\beta_U} \cdot [|\psi\rangle]_{\beta_U} \quad ,$$

portanto, a atuação da matriz de T sobre um ket é equivalente à multiplicação matriz-vetor coluna levando-se em consideração as bases previamente fixadas.

Considere que as bases β_U e β_V sejam ortonormais. Cada vetor $T|u_k\rangle$ pode ser escrito na base β_V da seguinte forma:

$$[T|u_k\rangle]_{\beta_V} = \begin{bmatrix} \langle v_0|Tu_k\rangle \\ \langle v_1|Tu_k\rangle \\ \vdots \\ \langle v_{m-1}|Tu_k\rangle \end{bmatrix}_{\beta_V} \quad ,$$

tendo em vista que a l -ésima entrada do vetor é o coeficiente da projeção de $|Tu_k\rangle$ na direção do l -ésimo vetor da base em V . Assim, a entrada de linha l e coluna k da matriz $[T]_{\beta_V}^{\beta_U}$ é $\langle v_l|Tu_k\rangle$, com $l = 0, \dots, m-1$ e $k = 0, \dots, n-1$ e consequentemente

$$[T]_{\beta_V}^{\beta_U} = \begin{bmatrix} \langle v_0|Tu_0\rangle & \langle v_0|Tu_1\rangle & \cdots & \langle v_0|Tu_{n-1}\rangle \\ \langle v_1|Tu_0\rangle & \langle v_1|Tu_1\rangle & \cdots & \langle v_1|Tu_{n-1}\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_{m-1}|Tu_0\rangle & \langle v_{m-1}|Tu_1\rangle & \cdots & \langle v_{m-1}|Tu_{n-1}\rangle \end{bmatrix} \quad .$$

No caso de um operador linear, tem-se $U = V$ ($m = n$), e é possível escolher a mesma base $\beta = \{|b_0\rangle, \dots, |b_{n-1}\rangle\}$ para o domínio e o contradomínio da transformação. Essa é uma situação bastante frequente, e a matriz associada ao operador linear é montada da seguinte forma: as colunas da matriz são os vetores $|Tb_k\rangle$ escritos como vetores coluna na base

β . Portanto:

$$[T]_{\beta} = [T]_{\beta}^{\beta} = \begin{bmatrix} \begin{array}{c} | \\ | Tb_0 \rangle \\ | \end{array} \quad \cdots \quad \begin{array}{c} | \\ | Tb_{n-1} \rangle \\ | \end{array} \end{bmatrix}_{\beta}$$

Se a base β for ortonormal, obtém-se que

$$[T]_{\beta} = \begin{bmatrix} \langle b_0 | Tb_0 \rangle & \langle b_0 | Tb_1 \rangle & \cdots & \langle b_0 | Tb_{n-1} \rangle \\ \langle b_1 | Tb_0 \rangle & \langle b_1 | Tb_1 \rangle & \cdots & \langle b_1 | Tb_{n-1} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle b_{n-1} | Tb_0 \rangle & \langle b_{n-1} | Tb_1 \rangle & \cdots & \langle b_{n-1} | Tb_{n-1} \rangle \end{bmatrix}.$$

Exemplo 1.20. Um operador linear A sobre um qubit pode ser escrito como uma matriz (na base computacional) 2×2 com coeficientes complexos da seguinte forma:

$$[A] = \begin{bmatrix} \begin{array}{c} | \\ A|0\rangle \\ | \end{array} & \begin{array}{c} | \\ A|1\rangle \\ | \end{array} \end{bmatrix} = \begin{bmatrix} \langle 0 | A | 0 \rangle & \langle 0 | A | 1 \rangle \\ \langle 1 | A | 0 \rangle & \langle 1 | A | 1 \rangle \end{bmatrix},$$

com $[\cdot]$ significando que os vetores em questão estão escritos como vetores coluna na base computacional. É frequente denotar a matriz do operador A pelo mesmo símbolo A , quando está implícito qual base é considerada.

Exemplo 1.21. A matriz da transformação linear do exemplo 1.19, na base computacional, é obtida escrevendo-se a ação de X sobre os vetores da base.

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

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

Em seguida, monta-se a matriz fazendo

$$X = \begin{bmatrix} \begin{array}{c} | \\ X|0\rangle \\ | \end{array} & \begin{array}{c} | \\ X|1\rangle \\ | \end{array} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Exemplo 1.22 (Matrizes de Pauli). As matrizes

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{e} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

são conhecidas como *matrizes de Pauli*. Essas são representações na base computacional dos operadores X , Y e Z . Usa-se, costumeiramente, a

mesma notação para se referir ao operador e à sua matriz na base computacional.

Em determinadas situações, a matriz identidade I também é chamada matriz de Pauli, e usa-se a notação alternativa

$$I = \sigma_0 \quad , \quad X = \sigma_x = \sigma_1 \quad , \quad Y = \sigma_y = \sigma_2 \quad \text{e} \quad Z = \sigma_z = \sigma_3 \quad .$$

1.3.6 Matriz da Composição de Transformações Lineares

A composição de transformações lineares $T: U \rightarrow V$ e $R: V \rightarrow W$ é a transformação linear denotada por $RT = R \circ T: U \rightarrow W$ e tal que $RT(|\psi\rangle) = R(T(|\psi\rangle))$ para todo $|\psi\rangle$. A matriz dessa transformação linear pode ser obtida pela multiplicação matricial das matrizes de R e de T :

$$[RT]_{\beta_W}^{\beta_U} = [R]_{\beta_W}^{\beta_V} \cdot [T]_{\beta_W}^{\beta_V} \quad ,$$

em que β_U , β_V e β_W são bases de U , V e W , respectivamente.

1.3.7 Mudança de Base

Para escrever a matriz de uma transformação linear $T: U \rightarrow V$ em novas bases β'_U e β'_V basta aplicar matrizes de mudança de base de maneira apropriada.

$$[T]_{\beta'_V}^{\beta'_U} = [I]_{\beta_U}^{\beta'_U} [T]_{\beta_V}^{\beta_U} [I]_{\beta'_V}^{\beta_V} \quad .$$

No caso de um operador linear $A: V \rightarrow V$, pode-se usar a mesma base nos espaços vetoriais do domínio e do contradomínio da função. A mudança de base nesse caso fica:

$$[A]_{\beta'_U}^{\beta'_U} = [I]_{\beta_U}^{\beta'_U} [A]_{\beta_U}^{\beta_U} [I]_{\beta'_U}^{\beta_U} = ([I]_{\beta'_U}^{\beta_U})^{-1} [A]_{\beta_U}^{\beta_U} ([I]_{\beta'_U}^{\beta_U}) \quad .$$

A transformação matricial $[A] \rightarrow [A]' = [M]^{-1}[A][M]$ é conhecida como *transformação de similaridade*. Duas transformações conectadas dessa forma são ditas *matrizes semelhantes*. As matrizes semelhantes são representantes de um mesmo operador linear escrito em bases diferentes.

Se as bases β_U e β'_U forem ortonormais, a fórmula para mudança de base fica:

$$[A]_{\beta'_U}^{\beta'_U} = [I]_{\beta_U}^{\beta'_U} [A]_{\beta_U}^{\beta_U} [I]_{\beta'_U}^{\beta_U} = ([I]_{\beta'_U}^{\beta_U})^\dagger [A]_{\beta_U}^{\beta_U} ([I]_{\beta'_U}^{\beta_U}) \quad ,$$

em que a operação simbolizada por \dagger é a transposição e conjugação da matriz. Essa operação será introduzida formalmente na seção 1.5.1.

Exemplo 1.23. Considere as bases \mathcal{I} e \mathcal{X} apresentadas no exemplo 1.2. A matriz de mudança de base de \mathcal{I} para \mathcal{X} e vice-versa é a matriz de Hadamard H , como visto no exemplo 1.5. O operador X , visto no exemplo 1.22, cuja matriz na base computacional é

$$X = [X]_{\mathcal{I}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

pode ser representado na base \mathcal{X} por

$$\begin{aligned} [X]_{\mathcal{X}} &= [I]_{\mathcal{X}}^{\mathcal{I}} [X]_{\mathcal{I}} [I]_{\mathcal{I}}^{\mathcal{X}} \\ &= H X H \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ &= Z . \end{aligned}$$

1.4 Autovalores, Autovetores e Decomposição Espectral

1.4.1 Autovalores e Autovetores

Seja A um operador linear, com matriz na base computacional também representada por A . Os *autovalores* de A são os números complexos λ que satisfazem

$$A|v\rangle = \lambda|v\rangle \quad \text{para algum } |v\rangle \neq 0 .$$

Os vetores não nulos $|v\rangle$ que satisfazem a equação acima são chamados *autovetores* de A associados ao autovalor λ .

Exemplo 1.24. O operador linear em 1 qubit Z definido pela matriz

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

possui:

- autovalor 1, pois o vetor não nulo $|0\rangle$ é tal que $Z|0\rangle = 1 \cdot |0\rangle$;
- autovalor -1 , pois o vetor não nulo $|1\rangle$ satisfaz $Z|1\rangle = -1 \cdot |1\rangle$.

1.4.2 Cálculo de Autovalores

A equação de autovalores $A|v\rangle = \lambda|v\rangle$ é equivalente a $(A - \lambda I)|v\rangle = 0$, com $|v\rangle \neq 0$, e isso é equivalente a dizer que a matriz de $A - \lambda I$ é singular. Por sua vez, isso equivale à equação

$$\det(A - \lambda I) = 0.$$

Com a equação acima, consegue-se encontrar os autovalores λ do operador A encontrando-se as raízes do *polinômio característico* $(A - \lambda I)$. Esse polinômio tem grau n e, como estamos buscando raízes nos números complexos, admite n raízes (pode acontecer que sejam repetidas). Dessa forma, todo operador admite um autovalor⁵.

Exemplo 1.25. Os autovalores da matriz

$$A = \begin{bmatrix} 0 & 2 \\ -1 & i \end{bmatrix}$$

podem ser obtidos por:

$$\begin{aligned} \det(A - \lambda I) &= 0 \\ \det \begin{bmatrix} 0 - \lambda & 2 \\ -1 & i - \lambda \end{bmatrix} &= 0 \\ -\lambda(i - \lambda) - 2 \cdot -1 &= 0 \\ \lambda^2 - i\lambda + 2 &= 0 \\ \lambda &= \frac{i \pm \sqrt{i^2 - 4 \cdot 1 \cdot 2}}{2 \cdot 1} \\ \lambda &= -i \quad \text{ou} \quad \lambda = 2i. \end{aligned}$$

Os autovalores podem ser números complexos. Como a matriz é 2×2 , foi obtido um polinômio característico de grau 2 e foram obtidas 2 raízes.

1.4.3 Cálculo de Autovetores

Uma vez descobertos os autovalores λ , retorna-se à equação $A|v\rangle = \lambda|v\rangle$, ou melhor, à equação

$$(A - \lambda I)|v\rangle = 0$$

para encontrar todos os autovetores $|v\rangle \neq 0$ satisfazendo essa equação. Como a matriz $A - \lambda I$ é singular (essa é a condição para se encontrar λ),

⁵Isso não é necessariamente válido para espaços vetoriais reais.

a equação em questão admite infinitas soluções $|v\rangle$, formando um sistema linear possível e indeterminado.

Em algumas situações é possível montar uma base para o espaço composta por autovetores do operador A . Há condições sobre o operador que revelam se é possível obter uma base ortonormal de autovetores. Isso será visto na seção 1.5. A obtenção de uma base de autovetores permite escrever a matriz A nessa base como uma matriz diagonal, o que se prova útil em diversas circunstâncias.

Exemplo 1.26. Dada a matriz

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

os autovalores e autovetores são encontrados a seguir.

Autovalores: Resolvendo $\det(X - \lambda I) = 0$, obtém-se:

$$\begin{aligned} \det(X - \lambda I) &= 0 \\ \det \begin{bmatrix} -\lambda & 1 \\ 1 & -\lambda \end{bmatrix} &= 0 \\ \lambda^2 - 1 &= 0 \\ \lambda &= \pm 1. \end{aligned}$$

Autovetores: Para cada autovalor λ , deve-se resolver

$$(X - \lambda I) |v\rangle = 0,$$

obtendo-se o vetor $|v\rangle$.

Para $\lambda = -1$:

$$\begin{aligned} (X - \lambda I) |v\rangle &= 0 \\ \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{cases} -a_0 + a_1 = 0 \\ a_0 - a_1 = 0 \end{cases} \end{aligned}$$

O sistema resultante, como esperado, é possível e indeterminado. Resolvendo o sistema, tem-se:

$$\begin{cases} a_0 = a_1 \\ a_1 \in \mathbb{C} \end{cases}.$$

Os autovetores associados ao autovalor $\lambda = -1$ são:

$$|v\rangle = \begin{bmatrix} a_1 \\ a_1 \end{bmatrix} = a_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{com } a_1 \in \mathbb{C}, a_1 \neq 0$$

O autoespaço associado a $\lambda = -1$ é o subespaço vetorial:

$$V_{-1} = \left\{ a_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} : a_1 \in \mathbb{C} \right\} = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}.$$

Para $\lambda = 1$:

$$\begin{aligned} (X - \lambda I) |v\rangle &= 0 \\ \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{cases} a_0 + a_1 = 0 \\ a_0 + a_1 = 0 \end{cases} \end{aligned}$$

O sistema é possível e indeterminado, e, resolvendo o sistema, tem-se:

$$\begin{cases} a_0 \in \mathbb{C} \\ a_1 = -a_0 \end{cases}.$$

Os autovetores associados ao autovalor $\lambda = 1$ são:

$$|v\rangle = \begin{bmatrix} a_0 \\ -a_0 \end{bmatrix} = a_0 \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad \text{com } a_0 \in \mathbb{C}, a_0 \neq 0$$

O autoespaço associado a $\lambda = 1$ é o subespaço vetorial:

$$V_1 = \left\{ a_0 \begin{bmatrix} 1 \\ -1 \end{bmatrix} : a_0 \in \mathbb{C} \right\} = \text{span} \left\{ \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\}.$$

1.4.4 Diagonalização de Operadores

Uma vez encontrados os autovalores λ_j e uma base de autovetores $\mathcal{V} = \{|v_j\rangle, j = 0, \dots, n-1\}$, com $|v_j\rangle$ associado ao autovalor λ_j , o operador linear A pode ser escrito na base \mathcal{V} como uma matriz diagonal. Defina as matrizes:

$$\begin{aligned} D &= \begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{bmatrix} \quad (\text{matriz diagonal dos autovalores}) \\ M &= \begin{bmatrix} | & & | \\ |v_0\rangle & \cdots & |v_{n-1}\rangle \\ | & & | \end{bmatrix} \quad (\text{matriz de mudança de base: } \mathcal{I} \rightarrow \mathcal{V}). \end{aligned}$$

A matriz de A na base \mathcal{V} é dada por:

$$\begin{aligned} [A]_{\mathcal{V}} &= \begin{bmatrix} [A|v_0\rangle]_{\mathcal{V}} & \cdots & [A|v_{n-1}\rangle]_{\mathcal{V}} \\ \vdots & & \vdots \\ [\lambda_0|v_0\rangle]_{\mathcal{V}} & \cdots & [\lambda_{n-1}|v_{n-1}\rangle]_{\mathcal{V}} \end{bmatrix} \\ &= \begin{bmatrix} \lambda_0 & & \\ & \ddots & \\ & & \lambda_{n-1} \end{bmatrix} = D \end{aligned}$$

A matriz M nada mais é que a matriz de mudança de base $M = [I]_{\mathcal{V}}^{\mathcal{Z}}$. Conforme a seção 1.1.3, pode-se escrever:

$$D = [A]_{\mathcal{V}} = [I]_{\mathcal{Z}}^{\mathcal{V}}[A][I]_{\mathcal{V}}^{\mathcal{Z}} = M^{-1}AM.$$

Ainda, se a base \mathcal{V} for ortonormal, conforme 1.2.4, pode-se escrever

$$D = M^{\dagger}AM.$$

Nessas expressões, usa-se A para denotar a matriz $[A]$ do operador A na base computacional, o que simplifica a notação quando não houver risco de confusão.

Portanto, de posse dos autovalores e de uma base ortonormal, é possível escrever o operador A como uma matriz diagonal.

O operador A também pode ser representado na notação do produto exterior da seguinte forma. Como \mathcal{V} forma uma base ortonormal, vale a relação de completude $I = \sum_k |v_k\rangle\langle v_k|$ para essa base. Aplicando-se essa relação a A , obtém-se

$$A = AI = \sum_k A|v_k\rangle\langle v_k| = \sum_k \lambda_k |v_k\rangle\langle v_k|.$$

Exemplo 1.27. No exemplo 1.26 foram calculados os autovalores e autovetores da matriz

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

obtendo-se:

- $\lambda = 1$, $|v\rangle = a \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ ($a \in \mathbb{C}, a \neq 0$)
- $\lambda = -1$, $|v\rangle = a \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ ($a \in \mathbb{C}, a \neq 0$)

Pretende-se extrair uma base ortonormal de autovetores para escrever X na forma diagonal. Nesse caso⁶, basta normalizar os autovetores obtidos.

Para $\lambda = -1$:

$$\left\| a \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\| = 1 \implies |a| \sqrt{1^2 + 1^2} = 1 \implies |a| = \frac{1}{\sqrt{2}}.$$

Há várias opções para a que satisfazem essa condição. Pode-se escolher apenas uma delas para fazer a diagonalização, por exemplo: $a = \frac{1}{\sqrt{2}}$. O autovetor normalizado é, portanto,

$$|v\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle.$$

Para $\lambda = 1$:

$$\left\| a \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\| = 1 \implies |a| \sqrt{1^2 + (-1)^2} = 1 \implies |a| = \frac{1}{\sqrt{2}}.$$

Do mesmo modo, pode-se escolher $a = \frac{1}{\sqrt{2}}$, e o autovetor normalizado é:

$$|v\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle.$$

Base ortonormal de autovetores:

$$\underbrace{|v_0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}}_{\lambda=1}, \quad \underbrace{|v_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}}_{\lambda=-1}$$

Diagonalização da matriz:

Matriz de X na forma diagonal:

$$X_D = \begin{bmatrix} \lambda_0 & \\ & \lambda_1 \end{bmatrix} = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix}$$

Matriz de mudança de base:

$$M = \begin{bmatrix} | & | \\ |v_0\rangle & |v_1\rangle \\ | & | \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

⁶Todos os autoespaços de dimensão 1.

Operador na forma diagonal: Na notação de produto exterior, tem-se:

$$X = \sum_k \lambda_k |v_k\rangle\langle v_k| = 1 \cdot |v_0\rangle\langle v_0| - 1 \cdot |v_1\rangle\langle v_1|$$

1.5 Tipos Especiais de Operadores

Nesta seção, alguns operadores com propriedades especiais serão apresentados. Os operadores normais, hermitianos, unitários, positivos e projetivos participam da base da Mecânica Quântica.

1.5.1 Operador Adjunto

Seja A um operador linear. Em Álgebra Linear, é possível demonstrar que existe um único operador linear, denotado por A^\dagger e chamado *operador adjunto* de A , que satisfaz a seguinte propriedade:

$$(OA1) \quad (|\phi\rangle, A|\psi\rangle) = (A^\dagger|\phi\rangle, |\psi\rangle).$$

Para qualquer base β , a matriz do operador A^\dagger está relacionada com a matriz de A por

$$(OA2) \quad [A^\dagger]_\beta = [A]_\beta^\dagger = ([A]_\beta^*)^T,$$

isto é, a matriz $[A]_\beta^\dagger$ é obtida de $[A]_\beta$ conjugando suas entradas e tomando a transposta.

Algumas propriedades da adjunta estão dispostas na seguinte lista:

- $(A^\dagger)^\dagger = A$ (Involução)
- $\left(\sum_k a_k A_k\right)^\dagger = \sum_k a_k^* A_k^\dagger$ (Antilinearidade)

Exemplo 1.28. A matriz adjunta de

$$A = \begin{bmatrix} 2 & 1+i \\ -1 & 5i \end{bmatrix}$$

é a matriz conjugada transposta

$$A^\dagger = \begin{bmatrix} 2 & -1 \\ 1-i & -5i \end{bmatrix}.$$

1.5.2 Operadores Normais

Um operador é dito *normal* se comuta com seu operador adjunto:

$$(ON) \quad A \cdot A^\dagger = A^\dagger \cdot A.$$

Exemplo 1.29. O operador definido pela matriz

$$A = \begin{bmatrix} 1+i & -1 \\ 1 & 1-i \end{bmatrix}$$

é normal, pois satisfaz $AA^\dagger = A^\dagger A$. De fato,

$$A^\dagger = \begin{bmatrix} 1-i & 1 \\ -1 & 1+i \end{bmatrix}$$

e tem-se que

$$AA^\dagger = \begin{bmatrix} 1+i & -1 \\ 1 & 1-i \end{bmatrix} \begin{bmatrix} 1-i & 1 \\ -1 & 1+i \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

$$A^\dagger A = \begin{bmatrix} 1-i & 1 \\ -1 & 1+i \end{bmatrix} \begin{bmatrix} 1+i & -1 \\ 1 & 1-i \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

A importância do operador normal decorre do seguinte teorema:

Teorema 1.30 (Teorema Espectral). *Um operador é normal se, e somente se, for diagonalizável.*

Dessa forma, basta checar se um operador é normal para se saber se ele admite uma base de autovetores e uma representação por matriz diagonal.

1.5.3 Operadores Hermitianos ou Autoadjuntos

Um operador H é dito *hermitiano*, ou *autoadjunto* se valer a seguinte propriedade:

$$(OH) \quad H^\dagger = H.$$

Exemplo 1.31. O operador dado por

$$A = \begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix}$$

é autoadjunto, pois

$$A^\dagger = \begin{bmatrix} 1 & -i \\ i & 1 \end{bmatrix} = A.$$

Os operadores hermitianos estão relacionados com a evolução no tempo de um sistema quântico fechado e com a medida de um observável.

Um operador hermitiano é, automaticamente, um operador normal, tendo em vista que $HH^\dagger = H^2 = H^\dagger H$. Conforme o teorema 1.30, todo operador hermitiano é, pois, diagonalizável. Além disso, há um teorema que permite tirar conclusões a respeito dos autovalores de uma matriz hermitiana.

Teorema 1.32 (Teorema Espectral para Matrizes Hermitianas). *Um operador normal é hermitiano se, e somente se, todos os seus autovalores são reais.*

1.5.4 Operadores Unitários

Um operador U é dito *unitário* se satisfizer alguma das condições equivalentes:

(OU1) $U^\dagger = U^{-1}$.

(OU2) As linhas ou as colunas de $[U]_\beta$ são vetores ortonormais em \mathbb{C}^n , para alguma base β .

(OU3) U é uma isometria, isto é, preserva o produto interno entre vetores (e em consequência, preserva também a distância entre vetores): $(U|\phi\rangle, U|\psi\rangle) = (|\phi\rangle, |\psi\rangle)$.

Exemplo 1.33. O operador definido pela matriz

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

é unitário, pois as colunas $|c_0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ e $|c_1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ são vetores ortonormais:

$$\begin{aligned} \||c_0\rangle\| &= \left\| \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\| = \frac{1}{\sqrt{2}} \sqrt{1^2 + 1^2} = 1 \\ \||c_1\rangle\| &= \left\| \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \right\| = \frac{1}{\sqrt{2}} \sqrt{1^2 + (-1)^2} = 1 \\ \langle c_0 | c_1 \rangle &= \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 1 - 1 = 0 \end{aligned}$$

O produto de dois operadores unitários é unitário:

$$(U_1 U_2)^\dagger = U_2^\dagger U_1^\dagger = U_2^{-1} U_1^{-1} = (U_1 U_2)^{-1}.$$

A condição de um operador ser unitário também implica normalidade, visto que $UU^\dagger = I = U^\dagger U$. De acordo com o teorema 1.30, todo operador unitário é, então, diagonalizável. Pode-se mostrar o seguinte teorema.

Teorema 1.34 (Teorema Espectral para Operadores Unitários). *Seja U uma matriz normal. U é uma matriz unitária se, e somente se, os seus autovalores são números complexos de módulo 1, logo exprimíveis na forma $\lambda = e^{i\theta}$ para algum $\theta \in \mathbb{R}$.*

1.5.5 Operadores Positivos

Um operador é dito *positivo* quando satisfaz a seguinte propriedade:

$$(\text{OPos}) \quad \langle \psi | P | \psi \rangle \geq 0, \quad \forall |\psi\rangle.$$

É possível demonstrar que um operador positivo é, automaticamente, hermitiano, e, portanto, todos os seus autovalores são reais. Além disso, a propriedade (OPos) é equivalente a dizer que todos os autovetores de P são números reais não-negativos $\lambda \geq 0$.

Diz-se que um operador é *positivo definido* quando satisfaz a condição seguinte, mais rigorosas que (OPos):

$$(\text{OPosDef}) \quad \langle \psi | P | \psi \rangle > 0, \quad \forall |\psi\rangle \neq 0$$

Essa condição é equivalente a afirmar que todos os autovalores de P são números reais positivos $\lambda > 0$.

Exemplo 1.35. O operador definido pela matriz

$$A = \begin{bmatrix} 1 & -2 \\ 0 & 2 \end{bmatrix}$$

é positivo. De fato, por se tratar de uma matriz triangular, os autovalores podem ser obtidos diretamente da diagonal: $\lambda = 1$ e $\lambda = 2$. Seus autovalores são todos não negativos, do que decorre que A é positivo. Como nenhum autovalor é nulo, o operador é também positivo definido.

1.5.6 Operadores de Projeção

Um *operador de projeção* é um operador P que satisfaz:

$$(\text{OProj}) \quad P^2 = P.$$

Os autovalores de P podem assumir os valores $\lambda = 0$ ou $\lambda = 1$. De fato, se λ é um autovalor com autovetor $|v\rangle \neq 0$ associado, tem-se $P|v\rangle = \lambda|v\rangle \implies \lambda|v\rangle = P|v\rangle = PP|v\rangle = \lambda P|v\rangle = \lambda^2|v\rangle$, logo, $(\lambda^2 - \lambda)|v\rangle = 0$

e, como $|v\rangle \neq 0$, tem-se $\lambda^2 - \lambda = 0$ e, por fim, $\lambda = 0$ ou $\lambda = 1$. Isso prova, em virtude dos teoremas 1.32 e 1.34, que todo operador de projeção é hermitiano e positivo.

Considere o subespaço vetorial de dimensão finita $W = P(V)$ (imagem de P). Seja $k = \dim W$ e tome uma base ortonormal $\{|v_0\rangle, \dots, |v_{k-1}\rangle\}$ de $P(V)$. Estendendo-se a $\{|v_0\rangle, \dots, |v_{k-1}\rangle, |v_k\rangle, \dots, |v_{n-1}\rangle\}$ base ortonormal do espaço V , é possível escrever o operador P como

$$P = \text{Proj}_W = \sum_{j=0}^{k-1} |v_j\rangle\langle v_j| .$$

O operador

$$Q = I - P = \text{Proj}_{W^\perp} = \sum_{j=k}^{n-1} |v_j\rangle\langle v_j|$$

é um operador de projeção no subespaço W^\perp .

Tem-se que $\text{Proj}_W + \text{Proj}_{W^\perp} = I$ e portanto, todo vetor $|\psi\rangle$ pode ser decomposto na soma de projeções em W e em W^\perp :

$$|\psi\rangle = \underbrace{\text{Proj}_W(|\psi\rangle)}_{\in W} + \underbrace{\text{Proj}_{W^\perp}(|\psi\rangle)}_{\in W^\perp} .$$

Exemplo 1.36. Os operadores no espaço de estados de 1 qubit

$$\begin{aligned} |0\rangle\langle 0| &= \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \\ |1\rangle\langle 1| &= \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} , \end{aligned}$$

são as projeções na direção dos vetores $|0\rangle$ e $|1\rangle$, respectivamente. Tem-se que

$$I = |0\rangle\langle 0| + |1\rangle\langle 1| ,$$

que é a relação de completude.

Se denotarmos por $W = \text{span}\{|0\rangle\}$ o subespaço gerado por $|0\rangle$, tem-se que $W^\perp = \text{span}\{|1\rangle\}$, $|0\rangle\langle 0| = \text{Proj}_W$, $|1\rangle\langle 1| = I - |0\rangle\langle 0| = \text{Proj}_{W^\perp}$ e qualquer vetor $|\psi\rangle$ pode ser escrito como

$$\begin{aligned} |\psi\rangle &= I|\psi\rangle = (|0\rangle\langle 0| + |1\rangle\langle 1|)|\psi\rangle \\ &= |0\rangle\langle 0||\psi\rangle + |1\rangle\langle 1||\psi\rangle \\ &= \text{Proj}_W|\psi\rangle + \text{Proj}_{W^\perp}|\psi\rangle . \end{aligned}$$

Se $|\psi\rangle = a|0\rangle + b|1\rangle$, então

$$\begin{aligned}\text{Proj}_W |\psi\rangle &= |0\rangle\langle 0| (a|0\rangle + b|1\rangle) = |0\rangle a \underbrace{\langle 0|0\rangle}_{=1} + |0\rangle b \underbrace{\langle 0|1\rangle}_{=0} = a|0\rangle \\ \text{Proj}_{W^\perp} |\psi\rangle &= |1\rangle\langle 1| (a|0\rangle + b|1\rangle) = |1\rangle a \underbrace{\langle 1|0\rangle}_{=0} + |1\rangle b \underbrace{\langle 1|1\rangle}_{=1} = b|1\rangle .\end{aligned}$$

1.5.7 Resumo

A figura abaixo traz um quadro resumo dos operadores especiais abordados neste capítulo.

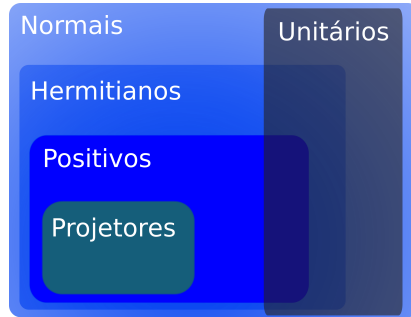


Figura 1.1: Relação entre os operadores especiais estudados neste capítulo.
Fonte: Slides de Álgebra Linear, UFSM [12]

Operador	Propriedade
Normal	$NN^\dagger = N^\dagger N$
Autoadjunto ou Hermitiano	$H^\dagger = H$
Unitário	$U^{-1} = U^\dagger$
Projutor	$P^2 = P$
Positivo	$\langle \psi P \psi \rangle \geq 0, \quad \forall \psi\rangle$
Positivo definido	$\langle \psi P \psi \rangle > 0, \quad \forall \psi\rangle \neq 0$

Tabela 1.1: Resumo das propriedades dos operadores especiais estudados neste capítulo.

1.6 Produto Tensorial

É possível compor dois espaços vetoriais para formar um terceiro espaço. Uma maneira de fazer isso é por meio do *produto tensorial*. Em Computação Quântica, essa construção é fundamental para se trabalhar com sistemas compostos por mais de um qubit. Um sistema de dois qubits será o produto tensorial de dois espaços \mathbb{C}^2 , que modelam um qubit.

1.6.1 Espaço Vetorial do Produto Tensorial

Dados dois espaços vetoriais V e W , com bases $\beta_V = \{|v_k\rangle\}$ e $\beta_W = \{|w_l\rangle\}$, o *produto tensorial* de V e W , denotado por $V \otimes W$, é definido como o espaço vetorial gerado pela base:

$$v_k \otimes w_l, \quad \begin{matrix} k = 0, \dots, \dim V - 1 \\ l = 0, \dots, \dim W - 1 \end{matrix}.$$

A dimensão do espaço vetorial do produto tensorial é, portanto,

$$\dim V \otimes W = \dim V \cdot \dim W.$$

O produto tensorial \otimes forma uma dupla ordenada com propriedades diferentes das do produto cartesiano. Essas propriedades, listadas abaixo, são chamadas conjuntamente de *bilinearidade*:

- Para todos $z \in \mathbb{C}$, $|v\rangle \in V$ e $|w\rangle \in W$,

$$z \cdot (|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle).$$

- Para todos $|v^1\rangle, |v^2\rangle \in V$ e $|w\rangle \in W$,

$$(|v^1\rangle + |v^2\rangle) \otimes |w\rangle = |v^1\rangle \otimes |w\rangle + |v^2\rangle \otimes |w\rangle.$$

- Para todos $|v\rangle \in V$ e $|w^1\rangle, |w^2\rangle \in W$,

$$|v\rangle \otimes (|w^1\rangle + |w^2\rangle) = |v\rangle \otimes |w^1\rangle + |v\rangle \otimes |w^2\rangle.$$

Um elemento genérico do espaço $V \otimes W$ é uma combinação linear dos vetores da base $|v_k\rangle \otimes |w_l\rangle$. Em geral, essa combinação não pode ser escrita da forma fatorada $|v\rangle \otimes |w\rangle$.

Exemplo 1.37. O sistema composto por 2 qubits é dado pelo produto tensorial de dois espaços vetoriais de 1 qubit (isto será visto com mais

detalhes na seção 2.1.5). Esse espaço vetorial é denotado por $\mathbb{C}^2 \otimes \mathbb{C}^2$. A base desse espaço é formada pelos 4 vetores

$$\begin{aligned} |00\rangle &= |0\rangle |0\rangle = |0\rangle \otimes |0\rangle \\ |01\rangle &= |0\rangle |1\rangle = |0\rangle \otimes |1\rangle \\ |10\rangle &= |1\rangle |0\rangle = |1\rangle \otimes |0\rangle \\ |11\rangle &= |1\rangle |1\rangle = |1\rangle \otimes |1\rangle , \end{aligned}$$

e as igualdades apresentadas acima apenas referem-se a notações alternativas e mais compactas. A ordem em que as entradas aparecem no produto tensorial é importante, de forma que $|01\rangle \neq |10\rangle$, por exemplo.

Um vetor qualquer $|\psi\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^2$ pode ser escrito como

$$|\psi\rangle = a |00\rangle + b |01\rangle + c |10\rangle + d |11\rangle ,$$

com $a, b, c, d \in \mathbb{C}$. Alguns exemplos de vetores pertencentes ao espaço em questão são:

$$\begin{aligned} &\frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ &|0\rangle \otimes (|0\rangle + 2|1\rangle) = |00\rangle + 2|01\rangle \\ &(5|1\rangle)|1\rangle = 5|11\rangle \\ &\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{i}{\sqrt{2}}|1\rangle\right)|0\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{i}{\sqrt{2}}|10\rangle . \end{aligned}$$

As igualdades acima são exemplos da bilinearidade do produto tensorial.

1.6.2 Comparação do Produto Tensorial com o Produto Cartesiano

O *produto cartesiano*, às vezes chamado *soma direta* é outra maneira de se compor dois espaços vetoriais em um espaço “maior”, denotado por $V \times W$ ou por $V \oplus W$. O produto cartesiano é formado por duplas $(|v\rangle, |w\rangle)$, com $|v\rangle \in V$ e $|w\rangle \in W$. Nesta subseção, a notação (\cdot, \cdot) refere-se a par ordenado em vez de produto interno como no restante do texto.

As diferenças entre essas duas operações estão dispostas no que segue: Outra diferença é que a soma, no produto cartesiano, é uma soma entrada a entrada

$$(|v^1\rangle, |w^1\rangle) + (|v^2\rangle, |w^2\rangle) = (|v^1\rangle + |v^2\rangle, |w^1\rangle + |w^2\rangle) ,$$

enquanto que a soma no produto tensorial, de modo geral, não se reduz

$$|v^1\rangle \otimes |w^1\rangle + |v^2\rangle \otimes |w^2\rangle ,$$

	Produto Tensorial	Produto Cartesiano ou Soma Direta
Notação:	$V \otimes W$	$V \times W = V \oplus W$
Base:	$ v_k\rangle \otimes w_l\rangle$	$(v_k\rangle, 0), (0, w_l\rangle)$
Dimensão:	$\dim V \otimes W = \dim V \cdot \dim W$	$\dim V \oplus W = \dim V + \dim W$

Tabela 1.2: Tabela comparativa entre produto tensorial e produto cartesiano (também chamado soma direta).

a não ser que, por exemplo, $|v^1\rangle = |v^2\rangle$, de modo que

$$|v^1\rangle \otimes |w^1\rangle + |v^1\rangle \otimes |w^2\rangle = |v^1\rangle \otimes (|w^1\rangle + |w^2\rangle) .$$

A multiplicação por escalar no produto cartesiano também é entrada a entrada

$$z \cdot (|v\rangle, |w\rangle) = (z|v\rangle, z|w\rangle) ,$$

enquanto que, no produto tensorial, o escalar pode ser incorporado a qualquer das duas entradas, mas deve ir para apenas uma delas

$$z \cdot (|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle) .$$

1.6.3 Produto Interno

Sejam V e W espaços de Hilbert, isto é, espaços munidos de produto interno. O produto tensorial $V \otimes W$ pode ser munido com um produto interno derivado dos produtos internos de V e de W . Defina:

$$\begin{aligned} \left(|v_k\rangle \otimes |w_l\rangle, |v_{k'}\rangle \otimes |w_{l'}\rangle \right) &= \left(|v_k\rangle, |v_{k'}\rangle \right)_V \cdot \left(|w_l\rangle, |w_{l'}\rangle \right)_W \\ &= \langle v_k | v_{k'} \rangle \cdot \langle w_l | w_{l'} \rangle = \delta_{k,k'} \delta_{l,l'} , \end{aligned} \quad (1.1)$$

estendendo a definição para elementos arbitrários do produto tensorial por linearidade na segunda entrada e antilinearidade na primeira.

Para dois vetores da forma $|v^1\rangle \otimes |w^1\rangle$ e $|v^2\rangle \otimes |w^2\rangle$ do produto tensorial, pode-se denotar

$$\left(|v^1\rangle \otimes |w^1\rangle, |v^2\rangle \otimes |w^2\rangle \right) = (\langle v^1 | \otimes \langle w^1 |) (|v^2\rangle \otimes |w^2\rangle) ,$$

e decorre da definição de produto interno que

$$(\langle v^1 | \otimes \langle w^1 |) (|v^2\rangle \otimes |w^2\rangle) = \langle v^1 | v^2 \rangle_V \cdot \langle w^1 | w^2 \rangle_W .$$

Exemplo 1.38. O espaço vetorial $\mathbb{C}^2 \otimes \mathbb{C}^2$ que descreve 2 qubits foi apresentado no exemplo 1.37. O produto interno nesse espaço é explicitado no que segue. Use os índices A e B para fazer referência à primeira e à segunda entrada tensorial, respectivamente.

O produto interno dos vetores da base é dado pela equação 1.1, que se traduz em

$$\begin{aligned} \langle jk|pq \rangle_{AB} &= (|jk\rangle, |pq\rangle)_{AB} \\ &= (|j\rangle, |p\rangle)_A \cdot (|k\rangle, |q\rangle)_B \\ &= \langle j|p \rangle_A \cdot \langle k|q \rangle_B \\ &= \delta_{j,p} \cdot \delta_{k,q} = \delta_{jk,pq} , \end{aligned}$$

em que $j, k, p, q = 0, 1$. Por exemplo,

$$\langle 01|01 \rangle = 1, \langle 11|10 \rangle = 0 \text{ e } \langle 10|01 \rangle = 0 .$$

Sejam

$$\begin{aligned} |\phi\rangle &= a_1 |00\rangle + b_1 |01\rangle + c_1 |10\rangle + d_1 |11\rangle \\ |\psi\rangle &= a_2 |00\rangle + b_2 |01\rangle + c_2 |10\rangle + d_2 |11\rangle . \end{aligned}$$

O produto interno de $|\phi\rangle$ com $|\psi\rangle$ é dado por

$$\begin{aligned} \langle \phi|\psi \rangle &= (a_1^* \langle 00| + b_1^* \langle 01| + c_1^* \langle 10| + d_1^* \langle 11|) (a_2 |00\rangle + b_2 |01\rangle + c_2 |10\rangle + d_2 |11\rangle) \\ &= a_1^* a_2 + b_1^* b_2 + c_1^* c_2 + d_1^* d_2 . \end{aligned}$$

A norma de $|\phi\rangle$ é dada por

$$\| |\phi\rangle \| = \sqrt{|a_1|^2 + |b_1|^2 + |c_1|^2 + |d_1|^2} .$$

Exemplos:

$$\begin{aligned} &\left(\frac{\langle 0| + \langle 1|}{\sqrt{2}} \right)_A \langle 1|_B |0\rangle_A \left(\frac{|0\rangle - i|1\rangle}{\sqrt{2}} \right)_B \\ &= \left(\frac{\langle 0|0\rangle + \langle 1|0\rangle}{\sqrt{2}} \right)_A \left(\frac{\langle 1|0\rangle - i\langle 1|1\rangle}{\sqrt{2}} \right)_B \\ &= \frac{1+0}{\sqrt{2}} \cdot \frac{0-i}{\sqrt{2}} = \frac{-i}{2} \end{aligned}$$

$$\| |01\rangle + i|10\rangle \| = \sqrt{|1|^2 + |i|^2} = \sqrt{2} .$$

1.6.4 Operadores

Sejam A um operador em V e B um operador em W . É possível definir um operador em $V \otimes W$, denotado por $A \otimes B$ de forma que

$$(A \otimes B)(|v\rangle \otimes |w\rangle) = A|v\rangle \otimes B|w\rangle .$$

Todo operador em $V \otimes W$ pode ser decomposto em uma combinação linear de operadores da forma apresentada anteriormente.

Pode-se mostrar que a composição, ou produto, de dois operadores $A \otimes B$ e $A' \otimes B'$ é dada por:

$$(A \otimes B)(A' \otimes B') = AA' \otimes BB' .$$

Exemplo 1.39. Sejam X e H operadores no espaço de 1 qubit dados por

$$\begin{aligned} X|0\rangle &= |1\rangle \\ X|1\rangle &= |0\rangle \end{aligned} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

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

Esses operadores foram utilizados em vários dentre os exemplos anteriores.

O operador $H \otimes X$ atua no espaço de 2 qubits como no exemplo abaixo.

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

Se forem usados rótulos 1 e 2 para as entradas tensoriais, a conta do exemplo acima poderia ser reescrita como

$$\begin{aligned} H_1 X_2 |0\rangle_1 (|0\rangle + i|1\rangle)_2 \\ &= H_1 |0\rangle_1 X_2 (|0\rangle + i|1\rangle)_2 \\ &= \left(\frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} \right) (|1\rangle_2 + i|0\rangle_2) \\ &= \frac{i}{\sqrt{2}}|00\rangle_{12} + \frac{1}{\sqrt{2}}|01\rangle_{12} + \frac{i}{\sqrt{2}}|10\rangle_{12} + \frac{1}{\sqrt{2}}|11\rangle_{12} . \end{aligned}$$

Mais detalhes sobre a notação encontram-se na seção 1.6.7

1.6.5 Produto de Kronecker

Até então, o conceito de produto tensorial foi apresentado de maneira abstrata. É possível abordar esse conceito de maneira matricial também. Fixadas bases para V e W , os vetores $|v\rangle$ e $|w\rangle$ desses espaços podem ser representados como vetores coluna. O vetor $|v\rangle \otimes |w\rangle$ também pode ser representado como vetor coluna fazendo-se o *produto de Kronecker*.

$$|v\rangle \otimes |w\rangle = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} \otimes \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{m-1} \end{bmatrix} = \begin{bmatrix} a_0 |w\rangle \\ a_1 |w\rangle \\ \vdots \\ a_{n-1} |w\rangle \end{bmatrix} = \begin{bmatrix} a_0 b_0 \\ a_0 b_1 \\ \vdots \\ a_0 b_{m-1} \\ a_1 b_0 \\ a_1 b_1 \\ \vdots \\ a_{n-1} b_{m-1} \end{bmatrix}$$

Essa operação é mais facilmente compreendida por meio de um exemplo.

Exemplo 1.40.

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \\ 2 \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \\ 8 \\ 10 \end{bmatrix}$$

O produto tensorial de operadores em V e W também pode ser obtido matricialmente por meio do produto de Kronecker.

$$A \otimes B = \begin{bmatrix} a_{0,0} & \cdots & a_{0,n-1} \\ a_{1,1} & & a_{1,n-1} \\ \vdots & & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,n-1} \end{bmatrix} \otimes B = \begin{bmatrix} a_{0,0}B & \cdots & a_{0,n-1}B \\ a_{1,1}B & & a_{1,n-1}B \\ \vdots & & \vdots \\ a_{n-1,0}B & \cdots & a_{n-1,n-1}B \end{bmatrix}$$

Exemplo 1.41.

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 1 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \\ 1 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & 0 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 3 & 4 \\ 1 & 2 & 0 & 0 \\ 3 & 4 & 0 & 0 \end{bmatrix}$$

O produto de Kronecker, em geral, não é comutativo.

Se $A \in M(m_A, n_A)$ e $B \in M(m_B, n_B)$, então a matriz $A \otimes B$ tem $m_A m_B$ linhas e $n_A n_B$ colunas, ou seja, $A \otimes B \in M(m_A m_B, n_A n_B)$. Uma forma geral de escrever o elemento j, k da matriz $A \otimes B$ é

$$(A \otimes B)_{j,k} = a_{\text{quoc}(j, m_B), \text{quoc}(k, n_B)} b_{\text{resto}(j, m_B), \text{resto}(k, n_B)} \quad ,$$

em que

$$\begin{aligned} j &= 0, 1, \dots, m_A m_B - 1 \\ k &= 0, 1, \dots, n_A n_B - 1 \end{aligned}$$

e, para x, y inteiros,

$$\begin{aligned} \text{quoc}(x, y) &\text{ é o quociente da divisão } x/y \\ \text{resto}(x, y) &\text{ é o resto da divisão } x/y . \end{aligned}$$

Exemplo 1.42. $A_{2 \times 2}$, $B_{3 \times 2}$. O elemento 4, 2 da matriz $A \otimes B$ pode ser obtido por:

$$(A \otimes B)_{4,2} = a_{\text{quoc}(4,3), \text{quoc}(2,2)} b_{\text{resto}(4,3), \text{resto}(2,2)} = a_{1,1} b_{1,0} .$$

A matriz $A \otimes B$ está representada abaixo, destacando-se o elemento 4, 2:

$$A \otimes B = \begin{bmatrix} a_{0,0}B & a_{0,1}B \\ a_{1,0}B & a_{1,1}B \end{bmatrix} = \begin{bmatrix} a_{0,0} \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \\ b_{2,0} & b_{2,1} \end{bmatrix} & a_{0,1} \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \\ b_{2,0} & b_{2,1} \end{bmatrix} \\ a_{1,0} \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \\ b_{2,0} & b_{2,1} \end{bmatrix} & \boxed{a_{1,1}} \begin{bmatrix} b_{0,0} & b_{0,1} \\ \boxed{b_{1,0}} & b_{1,1} \\ b_{2,0} & b_{2,1} \end{bmatrix} \end{bmatrix} .$$

1.6.6 Produto Tensorial de Vários Espaços Vetoriais

O produto tensorial de vários espaços vetoriais segue a mesma ideia do caso de dois espaços, apresentado anteriormente. A dimensão será o produto das dimensões de cada espaço. O produto tensorial é associativo, então não é necessário usar parênteses num produto tensorial de vários espaços. Não é possível comutar os fatores do produto tensorial.

1.6.7 Notação

Em situações práticas, costuma-se usar índices em cada fator do produto tensorial para evitar confusões. Também há variantes que deixam

a notação mais curta. Alguns comentários a respeito dessas variantes são abordados nesta seção. Para tratar disso, considere um exemplo em $V \otimes W \otimes U$.

Pode-se denotar um elemento da forma $|v\rangle \otimes |w\rangle \otimes |u\rangle$ omitindo-se o símbolo \otimes : $|v\rangle |w\rangle |u\rangle$. Nesse caso, deve-se tomar cuidado para não confundir essa justaposição com o produto matricial; nas situações de interesse, normalmente o produto matricial não será possível e há menos risco de confusão.

Costuma-se atribuir índices aos fatores: $|v\rangle \otimes |w\rangle \otimes |u\rangle = |v\rangle_V \otimes |w\rangle_W \otimes |u\rangle_U = |v\rangle_V |w\rangle_W |u\rangle_U$. Ainda, pode-se escrever esse elemento como: $|v\rangle_1 |w\rangle_2 |u\rangle_3$, ou $|v w u\rangle_{V W U}$, ou, ainda, qualquer notação equivalente, que evidencie a que espaço cada ket pertence. Com a identificação por índices, é possível até trocar a ordem em que são escritos; não deve ser confundido com a comutatividade dos fatores, que não é permitida em geral. Assim, pode-se escrever: $|u\rangle_3 |w\rangle_2 |v\rangle_1$.

Os bras também podem ser rotulados com índices. Por exemplo, escreve-se $(|v\rangle_1 |w\rangle_2 |u\rangle_3)^\dagger = {}_1\langle v| {}_2\langle w| {}_3\langle u|$, ou $|v w u\rangle_{V W U}^\dagger = {}_V\langle v| {}_W\langle w| {}_U\langle u|$.

Considere, agora, os operadores da forma $A \otimes B \otimes C$, com A , B e C operadores nos espaços V , W e U , respectivamente. É possível também atribuir índices nos operadores para lembrar em que espaço cada um deles atua. Por exemplo, pode-se denotar: $A_1 \otimes B_2 \otimes C_3$.

Há situações em que se deseja operar apenas em uma das entradas do produto tensorial. Assim, por exemplo, $A(|v\rangle \otimes |w\rangle \otimes |u\rangle) = A_1(|v\rangle \otimes |w\rangle \otimes |u\rangle)$. Formalmente, $A_1 = A \otimes I \otimes I$, nesse caso. O produto (ou composição) dos operadores $A_1 B_2 C_3$ significa $(A \otimes I \otimes I)(I \otimes B \otimes I)(I \otimes I \otimes C) = A I I \otimes I B I \otimes I C = A \otimes B \otimes C$. Dessa forma, com os índices, pode-se escrever $A_1 B_2 C_3 = A \otimes B \otimes C$ sem perigo de confusão com o produto matricial de A , B e C ; Pode-se até mesmo trocar a ordem de como são escritos: $B_2 A_1 C_3 = A \otimes B \otimes C$.

Outra notação bastante utilizada é quando se deseja realizar o produto tensorial entre n cópias de um vetor $|\psi\rangle$. Define-se

$$|\psi\rangle^{\otimes n} = \underbrace{|\psi\rangle \otimes |\psi\rangle}_{n \text{ vezes}}.$$

Essa notação pode ser utilizada para bras e para operadores: $\langle\psi|^{\otimes n}$, $A^{\otimes n}$. Também é possível usar uma notação análoga ao produtório para denotar, por exemplo,

$$\bigotimes_{i=1}^n A_i |0\rangle_i = A_1 |0\rangle_1 \otimes \dots \otimes A_n |0\rangle_n = A_1 \dots A_n |0 \dots 0\rangle.$$

Há, pois, diversas maneiras de se denotar os mesmos vetores ou operadores. Essa variedade é útil para permitir a escrita de expressões compactas

em diversas situações em que o produto tensorial aparece.

Capítulo 2

Introdução à Mecânica Quântica

A partir do final do século XIX e início do século XX, diversos fenômenos desafiavam a capacidade de explicação das teorias físicas vigentes. Dentre esses fenômenos, pode-se citar a radiação de corpo negro, o efeito fotoelétrico, as características ondulatórias do elétron em experimento de dupla fenda e o experimento de Stern-Gerlach. A Mecânica Quântica é produto das tentativas de se explicar esses fenômenos de modo compatível com o que se observa experimentalmente.

A apresentação da Mecânica Quântica, neste capítulo, é feita de maneira axiomática. Para manter o texto mais conciso e restrito ao objetivo de se estudar Computação Quântica, não são apresentadas motivações para os axiomas ou discussões a respeito dos experimentos. Uma discussão dos experimentos de fenda dupla e de Stern-Gerlach encontra-se em [3], seções 2.1 e 2.2, e em [15], seção 1.5.1.

As principais referências para este capítulo são os livros [15], seção 2.2 e [3], seção 2.4.

2.1 Postulados da Mecânica Quântica

2.1.1 Descrição de um Sistema Físico

O primeiro postulado prescreve como um sistema físico isolado pode ser descrito pela Mecânica Quântica. Um sistema *isolado* é um sistema no qual não ocorrem trocas de energia ou matéria com o ambiente.

Postulado 1. *Um **sistema físico isolado** é descrito por um espaço de Hilbert \mathcal{H} com escalares complexos. Esse espaço vetorial é chamado espaço de estados. Um estado do sistema é descrito pelo chamado vetor de estado, um vetor unitário no espaço de estados do sistema.*

A Mecânica Quântica não descreve qual espaço de Hilbert deve ser usado para descrever o sistema. Descobrir qual espaço de estados descreve adequadamente um dado sistema físico isolado é tarefa dos cientistas, que recorrem a teorias específicas ou propõem modelos para o sistema em questão. A Mecânica Quântica traz apenas a estrutura matemática e conceitual na qual espera-se que os sistemas físicos se encaixem.

No presente trabalho, apenas os espaços vetoriais de dimensão finita serão considerados. Nesse caso, um espaço de Hilbert é equivalente a um espaço vetorial com produto interno.

Exemplo 2.1 (Qubit). Um *qubit* é um sistema quântico descrito matematicamente por um espaço de Hilbert $\mathcal{H} = \mathbb{C}^2$ com $\dim \mathcal{H} = 2$. Qualquer base do espaço do sistema tem exatamente dois vetores. Uma base para o sistema é a base computacional $\{|0\rangle, |1\rangle\}$. O vetor nulo é costumeiramente denotado apenas por 0 para não ser confundido com o vetor da base $|0\rangle$.

Esse sistema é um análogo quântico ao bit clássico, um sistema que pode assumir os valores 0 ou 1 e que assume apenas um dos valores em cada instante. O sistema pode encontrar-se em qualquer estado $|\psi\rangle \in \mathcal{H}$ com $\| |\psi\rangle \| = 1$. Isto é, todo vetor $|\psi\rangle = a|0\rangle + b|1\rangle$ com $|a|^2 + |b|^2 = 1$ é capaz de descrever um estado do sistema.

Por exemplo, o qubit pode encontrar-se no estado $|0\rangle$ ou $|1\rangle$, de forma parecida com o bit clássico. A novidade agora é que ele pode se encontrar numa superposição desses estados, como $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, pois este é um vetor unitário em \mathcal{H} .

2.1.2 Evolução Temporal de um Sistema Físico

O estado de um sistema físico está sujeito a alterações com o tempo. A estrutura geral para essa evolução temporal do estado é dada pelo postulado a seguir.

Postulado 2. *A evolução temporal de um **sistema físico fechado** é dada por um operador linear unitário no seu espaço de estados \mathcal{H} . Em símbolos:*

$$|\psi_f\rangle = U |\psi_i\rangle .$$

Por ser um operador unitário, U preserva o tamanho dos vetores, de modo que o vetor $|\phi_f\rangle$ tem norma 1 e corresponde a um estado do sistema.

O enunciado do postulado 2 pode ser reescrito em termos de operadores Hermitianos.

Postulado 2'. *A evolução temporal de um sistema físico fechado é dada pela equação de Schroedinger*

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle ,$$

em que H é um operador Hermitiano no espaço de estados do sistema.

Neste postulado tem-se $\hbar = \frac{h}{2\pi}$, em que $h = 6,626 \cdot 10^{-34} \text{ J} \cdot \text{s}$ é a constante de Plank, determinada experimentalmente.

A equivalência entre esses dois postulados é dada por um teorema da Análise Funcional chamado Teorema de Stone¹. Para se ter uma ideia dessa equivalência entre os postulados, é mostrada uma argumentação não formal de que (2') implica (2) a seguir.

Afirmção. *O enunciado (2') implica (2).*

Prova da afirmação. Supõe-se que valha o enunciado em (2'). Seja H um operador hermitiano. Então $H = H^\dagger$. Considere inicialmente um intervalo de tempo dt infinitesimal. Pode-se escrever

$$\begin{aligned} |\psi(t+dt)\rangle &\cong |\psi(t)\rangle + \frac{d}{dt} |\psi(t)\rangle dt + \frac{1}{2!} \frac{d^2}{dt^2} |\psi(t)\rangle dt^2 + \dots \\ &\cong |\psi(t)\rangle + \frac{d}{dt} |\psi(t)\rangle dt , \end{aligned}$$

realizando a expansão em série de Taylor e retendo os termos até primeira ordem apenas. Pelo enunciado de (2'), tem-se que

$$\frac{d}{dt} |\psi(t)\rangle = -\frac{iH}{\hbar} |\psi(t)\rangle$$

Substituindo na equação anterior, tem-se que

$$|\psi(t+dt)\rangle \cong \left(I - \frac{iHdt}{\hbar} \right) |\psi(t)\rangle .$$

Definindo $U_{dt} = I - \frac{iHdt}{\hbar}$, tem-se que U_{dt} é unitária se desprezarmos os

¹Hall, B.C. (2013), Quantum Theory for Mathematicians, Graduate Texts in Mathematics, 267, Springer.

termos de ordem maior ou igual a 2 em dt . Efetivamente, tem-se:

$$\begin{aligned}
 U_{dt} U_{dt}^\dagger &= \left(I - \frac{iHdt}{\hbar} \right) \left(I + \frac{iH^\dagger dt}{\hbar} \right) \\
 &= I - \frac{iHdt}{\hbar} + \frac{iHdt}{\hbar} - \frac{i^2 H^2 dt^2}{\hbar^2} \\
 &\cong I \\
 U_{dt}^\dagger U_{dt} &= \left(I + \frac{iH^\dagger dt}{\hbar} \right) \left(I - \frac{iHdt}{\hbar} \right) \\
 &= I + \frac{iHdt}{\hbar} - \frac{iHdt}{\hbar} - \frac{i^2 H^2 dt^2}{\hbar^2} \\
 &\cong I .
 \end{aligned}$$

Para um intervalo de tempo finito de t_i a t_f , divida o intervalo de tempo em pedaços iguais $dt = \frac{t_f - t_i}{n}$, com $n \rightarrow \infty$. A evolução temporal entre t e $t + dt$ nesse intervalo de tempo é dada pelo operador U_{dt} . Dessa forma, tem-se que

$$t_f = t_i + n \cdot dt$$

e que

$$\begin{aligned}
 |\psi(t_f)\rangle &= U_{dt} |\psi(t_f - dt)\rangle \\
 &= U_{dt} U_{dt} |\psi(t_f - 2dt)\rangle \\
 &= \dots \\
 &= \underbrace{U_{dt} \dots U_{dt}}_{n \text{ vezes}} |\psi(t_i)\rangle
 \end{aligned}$$

Seja $U = (U_{dt})^n$. Com isso²,

$$\begin{aligned}
 U &= (U_{dt})^n = \left(I - \frac{iHdt}{\hbar} \right)^n \\
 &= \left(I - \frac{iH(t_f - t_i)}{n\hbar} \right)^n \\
 &\xrightarrow{n \rightarrow \infty} \exp \left(\frac{iH(t_f - t_i)}{\hbar} \right)
 \end{aligned}$$

Portanto tem-se

$$|\psi(t_f)\rangle = U |\psi(t_i)\rangle .$$

²Perceba que o limite $n \rightarrow \infty$ que surge é análogo ao limite fundamental

$$e^a = \lim_{n \rightarrow \infty} \left(1 - \frac{a}{n} \right)^n .$$

Além disso, U é um operador unitário pois é produto de operadores unitários, conforme seção 1.5.4. Assim, chegou-se ao enunciado (2). ┘

Exemplo 2.2. As matrizes de Pauli X , Y e Z apresentadas no exemplo 1.22 representam operadores lineares unitários em 1 qubit na base computacional e podem, portanto, representar a evolução temporal de um qubit.

Exemplo 2.3. A matriz de Hadamard H representa um operador linear unitário na base computacional, e pode, portanto, ser uma operação de evolução temporal de um sistema físico de 1 qubit.

Se o qubit se encontra no estado

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

após a aplicação de H , o estado final passa a ser

$$\begin{aligned} |\psi_f\rangle &= H \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle) \\ &= \frac{1}{\sqrt{2}}\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle + \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \\ &= |0\rangle \end{aligned}$$

2.1.3 Medidas em Sistemas Físicos

Um fenômeno muito particular da Mecânica Quântica é a medida de um observável em um sistema físico. Em geral, não é possível obter com certeza o resultado de uma medida. O que a teoria fornece são probabilidades de se obter cada possível resultado da medida.

Esse caráter probabilístico está atrelado aos fundamentos da Mecânica Quântica³. Isso significa que a Mecânica Quântica é uma teoria fundamentalmente probabilística.

Matematicamente, um observável é descrito por um operador Hermitiano e seus autovalores correspondem aos possíveis resultados da medida. Imediatamente após a medida, o estado é projetado no autoespaço associado ao valor obtido.

Postulado 3. *A cada observável físico associa-se um operador Hermitiano A no espaço de Hilbert do sistema. O operador A admite decomposição*

³Uma discussão interessante sobre o colapso da função de onda do sistema devido à medida é feita em [6], p. 2-5.

espectral e seus autovalores são reais (conforme seção 1.5.3). Seja

$$A = \sum_k a_k P_k$$

a sua decomposição espectral, com a_k autovalores distintos e P_k as projeções ortogonais nos autoespaços correspondentes.

Os valores a_k correspondem aos possíveis resultados da medida do observável A . Se o sistema está no estado $|\psi\rangle$ antes da medida, a probabilidade de o resultado ser a_k é dada por

$$p(a_k) = \|P_k |\psi\rangle\|^2.$$

O estado do sistema após a medida, dado que se obteve o resultado a_k , é dado por

$$\frac{P_k |\psi\rangle}{\|P_k |\psi\rangle\|}.$$

Caso o observável seja descrito por um operador A com todos os autovalores distintos, é possível escrever

$$A = \sum_k a_k |k\rangle\langle k|,$$

com os a_k s todos distintos e $P_k = |k\rangle\langle k|$. A probabilidade de se obter o valor a_k é

$$p(a_k) = |\langle k|\psi\rangle|^2,$$

e, dado que o resultado da medida tenha sido a_k , o estado do sistema passa a ser

$$\frac{\langle k|\psi\rangle}{|\langle k|\psi\rangle|} |k\rangle \sim |k\rangle \text{ (a menos de uma fase global)}.$$

Exemplo 2.4 (Qubit). Seja $|\psi\rangle = a|0\rangle + b|1\rangle$ o estado de um qubit, com $a, b \neq 0$. A medição desse qubit na base computacional, isto é, em relação ao observável

$$Z = 1 \cdot |0\rangle\langle 0| - 1 \cdot |1\rangle\langle 1|,$$

tem dois possíveis resultados mutuamente excludentes: 1 ou -1 . As probabilidades de se obter esses resultados são dadas a seguir.

O resultado 1 ocorre com probabilidade

$$p(1) = \||0\rangle\langle 0| |\psi\rangle\|^2 = \||0\rangle a\|^2 = |a|^2,$$

e o estado do sistema após a medida é

$$|\psi'\rangle = \frac{|0\rangle\langle 0| |\psi\rangle}{\||0\rangle\langle 0| |\psi\rangle\|} = \frac{a}{|a|} |0\rangle.$$

O resultado -1 ocorre com probabilidade

$$p(-1) = \| |1\rangle\langle 1| |\psi\rangle \|^2 = \| |1\rangle b \|^2 = |b|^2 ,$$

e o estado do sistema após a medida é

$$|\psi'\rangle = \frac{|1\rangle\langle 1| |\psi\rangle}{\| |1\rangle\langle 1| |\psi\rangle \|} = \frac{b}{|b|} |1\rangle .$$

Se o qubit estava no estado $|\psi\rangle = |0\rangle$, a probabilidade de se obter o resultado 1 na medição é $p(1) = 1 = 100\%$, e o estado após a medição é $|\psi'\rangle = |0\rangle$. Subsequentes medições continuarão fornecendo o resultado 1 e mantendo o estado final em $|0\rangle$.

Da mesma forma, se o qubit estava no estado $|\psi\rangle = |1\rangle$, a probabilidade de se obter o resultado -1 na medição é $p(-1) = 1 = 100\%$; o estado após a medição é $|\psi'\rangle = |1\rangle$ e subsequentes medições continuarão fornecendo o resultado -1 e mantendo o estado final em $|1\rangle$.

2.1.4 Valor esperado de um Observável

Um *ensemble* é uma coleção suficientemente grande de sistemas preparados no mesmo estado. Se realizarmos a medida de um observável em um ensemble, cada sistema tem um resultado possivelmente diferente, com probabilidades de acordo com o postulado 3. Pode-se extrair o valor esperado desse resultado, ou seja, a média dos resultados das medidas desses sistemas idênticos.

Definição 2.5 (Valor esperado de um observável). Seja A um observável, isto é, um operador hermitiano no espaço de estados do sistema. O *valor esperado* de A para esse ensemble preparado no estado $|\psi\rangle$, é dado pelo número real

$$\langle A \rangle = \langle \psi | A | \psi \rangle .$$

Observação 2.6. O valor esperado do observável depende do estado em que é preparado o ensemble. Por isso, pode-se usar algum rótulo para lembrar qual é o estado do ensemble:

$$\langle A \rangle = \langle A \rangle_\psi .$$

É comum também explicitar o estado do ensemble denotando o valor esperado simplesmente por $\langle \psi | A | \psi \rangle$.

Observação 2.7. O valor esperado dado na definição 2.5 corresponde à média dos resultados das medidas dos sistemas, como dito anteriormente. Sejam $A = \sum_k a_k P_k$ o referido observável e sua decomposição espectral.

Sejam $|\psi\rangle$ o estado de todos os sistemas do ensemble e N é o número de sistemas do ensemble. O valor esperado do observável A , denotado por $\langle A \rangle$, depende do estado $|\psi\rangle$ em que os sistemas se encontram e é dado pela média dos resultados das medidas. Como a probabilidade de se obter a_k na medição é $p(a_k) = \|P_k |\psi\rangle\|^2$, como informado no postulado 3, o número de sistemas com resultado a_k é $Np(a_k)$, e a média dos resultados é

$$\begin{aligned}
 \langle A \rangle &= \frac{1}{N} \sum_k a_k \cdot Np(a_k) \\
 &= \sum_k a_k p(a_k) \\
 &= \sum_k a_k \|P_k |\psi\rangle\|^2 \\
 &= \sum_k a_k \langle \psi | P_k^\dagger P_k | \psi \rangle \\
 &= \sum_k a_k \langle \psi | P_k | \psi \rangle \\
 &= \langle \psi | \sum_k a_k P_k | \psi \rangle \\
 &= \langle \psi | A | \psi \rangle ,
 \end{aligned}$$

lembrando que P_k , o operador projeção, é hermitiano ($P_k^\dagger = P_k$) e vale que $P_k^2 = P_k$.

2.1.5 Sistemas Compostos

O último postulado refere-se à união de dois ou mais sistemas em um único sistema composto. A ferramenta matemática que descreve adequadamente essa composição é o produto tensorial.

Postulado 4. *A composição de sistemas dados por $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_N$ é descrita pelo produto tensorial*

$$\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \dots \otimes \mathcal{H}_N .$$

Se o estado de cada sistema é $|\psi_1\rangle \in \mathcal{H}_1$, $|\psi_2\rangle \in \mathcal{H}_2$, \dots , $|\psi_N\rangle \in \mathcal{H}_N$, então o estado do sistema composto é dado pelo produto tensorial

$$|\psi_1\rangle \otimes \dots \otimes |\psi_N\rangle \in \mathcal{H}_1 \otimes \dots \otimes \mathcal{H}_N$$

dos estados dos sistemas componentes.

Exemplo 2.8 (Qubit). Suponha que se tenha dois qubits $\mathcal{H}_1 = \mathbb{C}^2$ e $\mathcal{H}_2 = \mathbb{C}^2$, preparados nos estados

$$\begin{aligned} |\psi_1\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ |\psi_2\rangle &= |0\rangle . \end{aligned}$$

A composição dos dois qubits é modelada pelo espaço de Hilbert $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 = \mathbb{C}^2 \otimes \mathbb{C}^2$, e o estado do sistema composto é descrito por

$$\begin{aligned} |\psi\rangle_{1,2} &= |\psi_1\rangle_1 \otimes |\psi_2\rangle_2 \\ &= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \right)_1 \otimes |0\rangle_2 \\ &= \frac{1}{\sqrt{2}}|0\rangle_1 \otimes |0\rangle_2 + \frac{1}{\sqrt{2}}|1\rangle_1 \otimes |0\rangle_2 \\ &= \frac{1}{\sqrt{2}}|0\rangle_1 |0\rangle_2 + \frac{1}{\sqrt{2}}|1\rangle_1 |0\rangle_2 \\ &= \frac{1}{\sqrt{2}}|00\rangle_{1,2} + \frac{1}{\sqrt{2}}|10\rangle_{1,2} \\ &= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle . \end{aligned}$$

As últimas 4 igualdades trazem diferentes notações para o mesmo vetor de estado. É comum omitir o símbolo \otimes e escrever as entradas tensoriais justapostas, ou ainda, dentro do mesmo ket. Para não haver confusão, pode-se usar índices para rotular os qubits (como os índices 1 e 2 usados acima). Caso não sejam utilizados os índices, a ordem em que os vetores são escritos na justaposição nos mostra a qual entrada tensorial cada vetor pertence, e um cuidado extra deve ser observado para não trocar a ordem das entradas. Por exemplo, $|10\rangle \neq |01\rangle$. Mais detalhes sobre a notação podem ser vistos na seção 1.6.7.

O produto tensorial permite que o sistema composto tenha uma propriedade muito importante: o *emaranhamento*.

Definição 2.9 (Estado Emaranhado / Estado Separável). Um sistema composto está em um *estado emaranhado* quando não é possível escrever o estado composto como um produto tensorial de estados individuais dos qubits:

$$|\psi\rangle \neq |\psi_1\rangle \otimes \dots \otimes |\psi_n\rangle .$$

Se o estado do sistema puder ser escrito dessa forma fatorada, diz-se que é um *estado separável*, ou *fatorável*, ou ainda, *estado produto*.

Para um sistema composto por 2 qubits, um estado $|\psi\rangle$ é separável se e somente se puder ser escrito na forma

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

Exemplo 2.10 (Qubit). Num sistema composto por 2 qubits, o estado

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

é um estado emaranhado. De fato, se fosse um estado separável, poderia ser escrito como

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = ac|00\rangle + ad|01\rangle + bc|10\rangle + bd|11\rangle .$$

Se isso ocorresse, então

$$\begin{cases} ac = \frac{1}{\sqrt{2}} & (1) \\ ad = 0 & (2) \\ bc = 0 & (3) \\ bd = \frac{1}{\sqrt{2}} & (4) \end{cases}$$

O sistema não tem solução, pois as equações (1) e (4) implicam $a, b, c, d \neq 0$ e a equação (2) implica $a = 0$ ou $d = 0$, levando a uma contradição.

Se tentarmos medir apenas o primeiro qubit de $|\beta_{00}\rangle_{1,2}$ (na base computacional), teremos o resultado $|0\rangle_1$ com probabilidade $\frac{1}{2}$ e $|1\rangle_1$ com probabilidade também $\frac{1}{2}$. Se o resultado da medição for $|0\rangle_1$, então o estado do sistema composto será $|00\rangle_{1,2}$, e portanto o segundo qubit também colapsará para o estado $|0\rangle_2$. Da mesma forma, se o resultado da medição for $|1\rangle_1$, o estado do sistema composto ficará $|11\rangle_{1,2}$, e o segundo qubit colapsará para o estado $|1\rangle_2$. Medir o estado do qubit 1 afeta o qubit 2, se ambos estiverem emaranhados, mesmo que estejam muito distantes!

2.2 Matrizes de Pauli

As matrizes de Pauli são algumas das ferramentas mais corriqueiras em Computação Quântica. Essas matrizes já foram introduzidas no capítulo de Álgebra Linear, no exemplo 1.22, e algumas propriedades já foram abordadas nos exemplos 1.24, 1.26 e 1.27. As principais informações a respeito dessas matrizes são reunidas nesta seção por conveniência.

2.2.1 Definição

As matrizes de Pauli são reproduzidas na tabela a seguir.

Notação	Ação na base $ 0\rangle, 1\rangle$	Matriz na base $ 0\rangle, 1\rangle$
$X = \sigma_x = \sigma_1$	$X 0\rangle = 1\rangle$ $X 1\rangle = 0\rangle$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
$Y = \sigma_y = \sigma_2$	$Y 0\rangle = i 1\rangle$ $Y 1\rangle = -i 0\rangle$	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
$Z = \sigma_z = \sigma_3$	$Z 0\rangle = 0\rangle$ $Z 1\rangle = - 1\rangle$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

2.2.2 Propriedades

As matrizes de Pauli são *hermitianas* e *unitárias*. Para verificar isso, basta perceber que cada matriz é igual à sua adjunta (conjugada transposta) e que suas colunas são ortonormais. Portanto vale que

$$\begin{aligned} X^\dagger &= X & X^\dagger &= X^{-1} \\ Y^\dagger &= Y & Y^\dagger &= Y^{-1} \\ Z^\dagger &= Z & Z^\dagger &= Z^{-1} . \end{aligned}$$

Dessas igualdades segue que cada matriz de Pauli é igual à sua inversa, e que portanto, o seu quadrado é a matriz identidade.

$$\begin{aligned} X^2 &= I \\ Y^2 &= I \\ Z^2 &= I . \end{aligned}$$

Há outras identidades envolvendo o produto de duas matrizes de Pauli diferentes, e que podem ser obtidas por cálculo direto.

$$\begin{aligned} XY &= iZ & YX &= -iZ \\ YZ &= iX & ZY &= -iX \\ ZX &= iY & XZ &= -iY . \end{aligned}$$

As identidades da coluna à direita podem ser obtidas fazendo-se o hermitiano das igualdades da coluna à esquerda.

2.2.3 Autovalores, Autovetores e Diagonalização

Por serem matrizes hermitianas e unitárias, são matrizes normais (isto é, comutam com sua adjunta). São, pois, diagonalizáveis pelo teorema espectral 1.30, têm autovalores reais por serem hermitianas e seus autovalores devem ter módulo 1 por serem matrizes unitárias. Um cálculo explícito fornece os autovalores 1 e -1 . Cada matriz possui uma base de autovetores diferente, denotada por \mathcal{X} , \mathcal{Y} e \mathcal{Z} e apresentadas no exemplo 1.2. O cálculo dos autovalores e autovetores segue as diretrizes do exemplo 1.26 e 1.27, em que são encontrados os autovalores, autovetores e a forma diagonal da matriz X .

Matriz	Autovalores	Base de Autovetores	Forma diagonal
X	1, -1	$\mathcal{X} = \{ +\rangle, -\rangle \}$	$X = +\rangle\langle + - -\rangle\langle - $
Y	1, -1	$\mathcal{Y} = \{ +i\rangle, -i\rangle \}$	$Y = +i\rangle\langle +i - -i\rangle\langle -i $
Z	1, -1	$\mathcal{Z} = \{ 0\rangle, 1\rangle \} = \mathcal{I}$	$Z = 0\rangle\langle 0 - 1\rangle\langle 1 $

Tabela 2.1: Autovalores, base de autovetores e forma diagonal dos operadores X , Y e Z de Pauli.

Os vetores $|+\rangle$, $|-\rangle$, $|+i\rangle$ e $|-i\rangle$ são dados por:

$$\begin{aligned} |+\rangle &= \frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle & |+i\rangle &= \frac{1}{\sqrt{2}} |0\rangle + i \frac{1}{\sqrt{2}} |1\rangle \\ |-\rangle &= \frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle & |-i\rangle &= \frac{1}{\sqrt{2}} |0\rangle - i \frac{1}{\sqrt{2}} |1\rangle . \end{aligned}$$

2.3 Estados de Bell

Os *estados de Bell* são estados emaranhados que formam uma base ortonormal \mathcal{B} para o espaço de estados de 2 qubits. Um deles já foi visto no exemplo 2.10. Os 4 estados de Bell estão dispostos a seguir.

$$\begin{aligned} |\beta_{00}\rangle &= |\Phi^+\rangle = \frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle \\ |\beta_{01}\rangle &= |\Phi^-\rangle = \frac{1}{\sqrt{2}} |00\rangle - \frac{1}{\sqrt{2}} |11\rangle \\ |\beta_{10}\rangle &= |\Psi^+\rangle = \frac{1}{\sqrt{2}} |01\rangle + \frac{1}{\sqrt{2}} |10\rangle \\ |\beta_{11}\rangle &= |\Psi^-\rangle = \frac{1}{\sqrt{2}} |01\rangle - \frac{1}{\sqrt{2}} |10\rangle \end{aligned}$$

Na literatura encontram-se os dois tipos de notação.

Proposição 2.11. *Os estados de Bell formam uma base ortonormal para o espaço de estados de 2 qubits $\mathbb{C}^2 \otimes \mathbb{C}^2$.*

Demonstração.

\mathcal{B} é base de $\mathbb{C}^2 \otimes \mathbb{C}^2$:

A dimensão de $\mathbb{C}^2 \otimes \mathbb{C}^2$ é 4, portanto basta mostrar que os 4 estados de Bell são LI. Seja

$$a_0 |\beta_{00}\rangle + a_1 |\beta_{01}\rangle + a_2 |\beta_{10}\rangle + a_3 |\beta_{11}\rangle = 0$$

uma combinação linear nula dos elementos de \mathcal{B} . Tem-se que:

$$\begin{aligned} a_0 |\beta_{00}\rangle + a_1 |\beta_{01}\rangle + a_2 |\beta_{10}\rangle + a_3 |\beta_{11}\rangle &= 0 \\ a_0 \frac{|00\rangle + |11\rangle}{\sqrt{2}} + a_1 \frac{|00\rangle - |11\rangle}{\sqrt{2}} + a_2 \frac{|01\rangle + |10\rangle}{\sqrt{2}} + a_3 \frac{|01\rangle - |10\rangle}{\sqrt{2}} &= 0 \\ \frac{a_0 + a_1}{\sqrt{2}} |00\rangle + \frac{a_2 + a_3}{\sqrt{2}} |01\rangle + \frac{a_2 - a_3}{\sqrt{2}} |10\rangle + \frac{a_0 - a_1}{\sqrt{2}} |11\rangle &= 0 \end{aligned}$$

Portanto

$$\begin{cases} a_0 + a_1 = 0 \\ a_0 - a_1 = 0 \\ a_2 + a_3 = 0 \\ a_2 - a_3 = 0 \end{cases} \implies \begin{cases} a_0 = 0 \\ a_1 = 0 \\ a_2 = 0 \\ a_3 = 0 \end{cases},$$

e como a única solução para os coeficientes é a solução nula, os vetores são LI. Formam, em consequência, uma base do espaço de 2 qubits.

\mathcal{B} é ortonormal:

Os vetores têm norma 1 pois

$$\begin{aligned} \langle \beta_{00} | \beta_{00} \rangle &= \frac{\langle 00 | + \langle 11 |}{\sqrt{2}} \frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1 + 0 + 0 + 1}{2} = 1 \\ \langle \beta_{01} | \beta_{01} \rangle &= \frac{\langle 00 | - \langle 11 |}{\sqrt{2}} \frac{|00\rangle - |11\rangle}{\sqrt{2}} = \frac{1 - 0 - 0 + 1}{2} = 1 \\ \langle \beta_{10} | \beta_{10} \rangle &= \frac{\langle 01 | + \langle 10 |}{\sqrt{2}} \frac{|01\rangle + |10\rangle}{\sqrt{2}} = \frac{1 + 0 + 0 + 1}{2} = 1 \\ \langle \beta_{11} | \beta_{11} \rangle &= \frac{\langle 01 | - \langle 10 |}{\sqrt{2}} \frac{|01\rangle - |10\rangle}{\sqrt{2}} = \frac{1 - 0 - 0 + 1}{2} = 1, \end{aligned}$$

e são ortogonais porque

$$\begin{aligned}
 \langle \beta_{00} | \beta_{01} \rangle &= \frac{\langle 00 | + \langle 11 |}{\sqrt{2}} \frac{|00\rangle - |11\rangle}{\sqrt{2}} = \frac{1 - 0 + 0 - 1}{2} = 0 \\
 \langle \beta_{00} | \beta_{10} \rangle &= \frac{\langle 00 | + \langle 11 |}{\sqrt{2}} \frac{|01\rangle + |10\rangle}{\sqrt{2}} = \frac{0 + 0 + 0 + 0}{2} = 0 \\
 \langle \beta_{00} | \beta_{11} \rangle &= \frac{\langle 00 | + \langle 11 |}{\sqrt{2}} \frac{|01\rangle - |10\rangle}{\sqrt{2}} = \frac{0 - 0 + 0 - 0}{2} = 0 \\
 \langle \beta_{01} | \beta_{10} \rangle &= \frac{\langle 00 | - \langle 11 |}{\sqrt{2}} \frac{|01\rangle + |10\rangle}{\sqrt{2}} = \frac{0 + 0 - 0 - 0}{2} = 0 \\
 \langle \beta_{01} | \beta_{11} \rangle &= \frac{\langle 00 | - \langle 11 |}{\sqrt{2}} \frac{|01\rangle - |10\rangle}{\sqrt{2}} = \frac{0 - 0 - 0 + 0}{2} = 0 \\
 \langle \beta_{10} | \beta_{11} \rangle &= \frac{\langle 01 | + \langle 10 |}{\sqrt{2}} \frac{|01\rangle - |10\rangle}{\sqrt{2}} = \frac{1 - 0 + 0 - 1}{2} = 0 .
 \end{aligned}$$

Logo a base é ortonormal. \square

Proposição 2.12. *Os estados de Bell são emaranhados (isto é, não podem ser escritos como o produto tensorial de estados de 1 qubit).*

Demonstração. As contas são análogas às do exemplo 2.10, que mostra que o estado $|\beta_{00}\rangle$ é emaranhado. \square

Os estados de Bell são muito utilizados pelo fato de formarem uma base de estados emaranhados. Algumas aplicações mais elementares ocorrem no circuito de teletransporte e na codificação superdensa, que serão vistos no capítulo 6, seções 6.1 e 6.2, respectivamente.

Computação Quântica

A *Computação Quântica* é um novo paradigma de computação em que utilizam-se sistemas quânticos – os *qubits*, análogos dos bits clássicos – para se realizar processamento de informação. Lançam-se mão de alguns recursos não existentes na Computação Clássica, como *superposição* e *emaranhamento*.

Há uma variedade de modelos de Computação Quântica, dentre os quais pode-se citar:

- **Computação Quântica de Circuitos** – computação realizada com portas lógicas quânticas, análogas às portas lógicas dos sistemas digitais clássicos;
- **Computação Quântica Adiabática** – o sistema é preparado no estado fundamental e sofre a ação de um hamiltoniano que depende continuamente do tempo e que é projetado de forma que o estado fundamental, após a aplicação do hamiltoniano, contenha a solução do problema codificada ([7], p.23);
- **Máquina de Turing Quântica** – uma versão quântica da Máquina de Turing Clássica;
- **Caminhada Aleatória Quântica** – versão quântica da caminhada aleatória clássica; é possível realizar computação quântica universal com esse modelo de computação.

O presente trabalho aborda apenas a Computação Quântica de Circuitos.

3.1 Computação Quântica de Circuitos

Este modelo de Computação Quântica guarda analogia com a Computação Clássica com Portas Lógicas, apresentada no apêndice A (principalmente seção A.3). A informação é codificada em qubits, e o processamento é feito por evolução temporal do sistema segundo operações unitárias (conforme postulado 2).

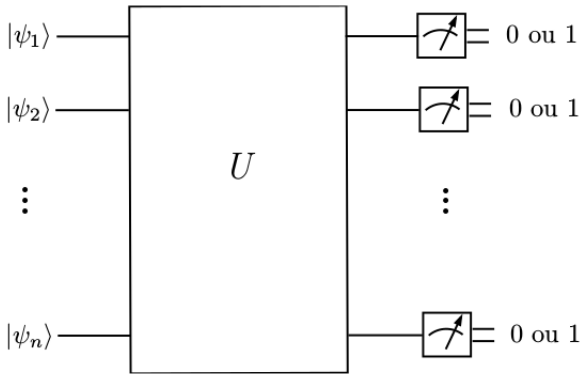


Figura 3.1: Esquema geral de um computador quântico no modelo de circuitos. Os qubits são preparados em estados iniciais e são submetidos a operações unitárias simbolizadas pela caixa de rótulo U . Após a sequência de operações, os qubits são lidos (ocorre a medição dos qubits na base computacional), fornecendo uma sequência de bits clássicos com o resultado da computação. Fonte: [17], p. 24.

Número de entradas e saídas

Os circuitos quânticos devem ter o mesmo número de entradas e saídas, pois qubits não são introduzidos ou removidos durante o processamento e as portas lógicas quânticas são operadores, o que significa que preservam o número de qubits em que atuam. Diferentemente do que ocorre com os circuitos clássicos, nos quais há portas com diferentes números de bits à entrada e à saída (como as portas AND e NOT, por exemplo).

Reversibilidade da Computação Quântica

Outra diferença é os circuitos quânticos são essencialmente *reversíveis*, ou seja, existe um circuito quântico inverso que consegue retornar as entradas originais do circuito a partir das saídas do mesmo. Isso ocorre porque

o processamento se dá por operadores unitários, que são reversíveis, com inversos também dados por operadores unitários. Apesar de a maior parte da Computação Quântica ser reversível, há uma etapa irreversível: a medição dos qubits.

Nos circuitos clássicos, a maioria das portas lógicas não são reversíveis. A porta AND, por exemplo, fornece resultado 1 se ambas as entradas forem 1 e fornece 0 caso contrário; no caso de a saída ser 0, não sabemos qual/quais das entradas é 0.

Não há um impedimento absoluto para a Computação Clássica ser reversível. De fato, há investigações relacionadas a circuitos clássicos reversíveis, constituídos apenas por portas lógicas reversíveis.

Há uma expectativa de que a computação reversível seja mais eficiente em termos energéticos que a computação irreversível. Isso se deve ao *princípio de Landauer*, que diz que o apagamento de 1 bit de informação está associado a uma dissipação de energia para o ambiente de, no mínimo, $kT \ln 2$ (ver referência [3], p. 37). Em tese, se forem evitados os apagamentos de informação, a computação poderia ser feita sem gasto energético apreciável. Uma discussão interessante sobre o consumo energético e computação pode ser encontrada em [15], na seção 3.2.5, e em [3], seção 1.5.

3.2 O Qubit

3.2.1 Descrição Matemática do Qubit

O qubit é um sistema físico que pode ser descrito por um espaço de Hilbert de dimensão 2. O qubit já foi apresentado no capítulo anterior, exemplo 2.1. Os estados da base canônica são rotulados por $|0\rangle$ e $|1\rangle$, e um estado geral para o qubit é o vetor unitário

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

em que $|a|^2 + |b|^2 = 1$ e $a, b \in \mathbb{C}$.

3.2.2 Fase Relativa e Fase Global

Fase Global

Chama-se *fase global* uma fase complexa multiplicando o estado de um qubit: $e^{i\alpha}|\psi\rangle$. Dois estados $|\psi_1\rangle$ e $|\psi_2\rangle = e^{i\alpha}|\psi_1\rangle$, iguais a menos de uma fase global, não podem ser distinguidos fisicamente. De fato, dado um observável $A = \sum_k a_k P_k$, a probabilidade de o resultado de uma medida

ser a_k , nos dois casos, é igual:

$$p_2(a_k) = \|P_k e^{i\alpha} |\psi_1\rangle\|^2 = |e^{i\alpha}|^2 \| |\psi_1\rangle \|^2 = p_1(a_k) .$$

A evolução temporal nos dois estados também é idêntica, a menos do fator $e^{i\alpha}$, devido à linearidade dos operadores de evolução:

$$U |\psi_2\rangle = U e^{i\alpha} |\psi_1\rangle = e^{i\alpha} U |\psi_1\rangle ,$$

e uma subsequente medição não conseguiria distinguir esses dois estados que diferem apenas por uma fase global. Na formação de um sistema composto, os vetores $|\psi_1\rangle$ e $|\psi_2\rangle = e^{i\alpha} |\psi_1\rangle$ produzem resultados idênticos, a menos da fase global α , dada a multilinearidade do produto tensorial:

$$|\psi_2\rangle \otimes |\phi\rangle = (e^{i\alpha} |\psi_1\rangle) \otimes |\phi\rangle = e^{i\alpha} (|\psi_1\rangle \otimes |\phi\rangle) .$$

Do mesmo modo, uma medida posterior não conseguiria distinguir esses dois estados.

Dessa forma, a fase global não tem relevância física, e um sistema descrito por um vetor de estado $|\psi\rangle$ também pode ser descrito pelo vetor $e^{i\alpha} |\psi\rangle$.

Fase Relativa

A *fase relativa* em um qubit é a diferença de fase entre os coeficientes que multiplicam o $|1\rangle$. Por exemplo, os vetores

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

têm mesmo coeficiente multiplicando $|0\rangle$ e diferem apenas por um fator $-1 = e^{i\pi}$ multiplicando $|1\rangle$, isto é, por uma fase relativa de π .

De forma geral, os estados

$$|\psi_1\rangle = a|0\rangle + b|1\rangle \quad \text{e} \quad |\psi_2\rangle = a|0\rangle + b e^{i\varphi} |1\rangle$$

diferem por fase relativa φ . Esses estados apresentam mesmas probabilidades em uma medida na base computacional:

$$p_2(|0\rangle) = |a|^2 = p_1(|0\rangle)$$

$$p_2(|1\rangle) = |b e^{i\varphi}|^2 = |b|^2 |e^{i\varphi}|^2 = |b|^2 = p_1(|1\rangle) ,$$

no entanto, em bases diferentes, podem apresentar probabilidades diferentes. Exemplificando, os estados $|+\rangle$ e $|-\rangle$ diferem por uma fase relativa,

no entanto formam uma base. E a medida nessa base fornece resultados distintos. A evolução por transformações unitárias também apresenta resultados diferentes. Por exemplo, a aplicação do operador unitário H fornece $H|+\rangle = |0\rangle$ e $H|-\rangle = |1\rangle$.

Assim, ao contrário da fase global, a fase relativa apresenta relevância física.

3.2.3 Representação de um Qubit na Esfera de Bloch

A Esfera de Bloch

O estado $|\psi\rangle = a|0\rangle + b|1\rangle$ de um qubit pode ser reescrito, a menos de uma fase global, como

$$|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle, \quad \theta \in [0, \pi], \varphi \in [0, 2\pi).$$

Isso será justificado na seção 3.2.4. Pode-se multiplicar o estado por uma fase global para que o termo multiplicando $|0\rangle$ seja real e positivo. Fazendo-se essa identificação em relação à fase global, tem-se o estado de um qubit descrito por dois parâmetros θ e φ . Utilizando esses dois parâmetros no sistema de coordenadas esféricas, pode-se corresponder os estados de um qubit com os pontos na superfície de uma esfera de raio unitário, a chamada *Esfera de Bloch*.

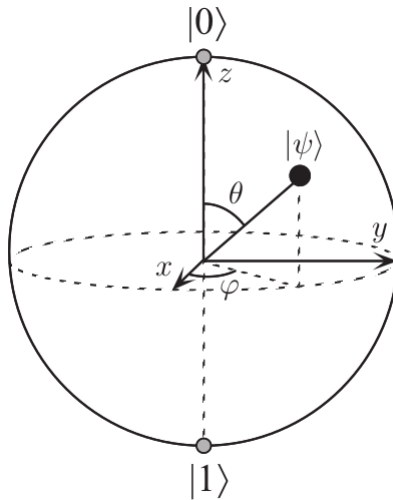


Figura 3.2: Representação de um qubit na esfera de Bloch. Fonte: [15], p.15

Pontos na Esfera de Bloch

Os polos norte da esfera corresponde ao estado $|0\rangle$ e o polo sul, ao $|1\rangle$. No equador, situam-se os estados da forma $\frac{1}{\sqrt{2}}|0\rangle + \frac{e^{i\varphi}}{\sqrt{2}}|1\rangle$, isto é, superposições dos estados $|0\rangle$ e $|1\rangle$ com o mesmo peso e com alguma fase relativa.

Ponto da esfera de Bloch			Estado $ \psi\rangle$
$\hat{x} = (1, 0, 0)$	$\theta = \pi/2$	$\varphi = 0$	$ +\rangle$
$-\hat{x} = (-1, 0, 0)$	$\theta = \pi/2$	$\varphi = \pi$	$ -\rangle$
$\hat{y} = (0, 1, 0)$	$\theta = \pi/2$	$\varphi = \pi/2$	$ +i\rangle$
$-\hat{y} = (0, -1, 0)$	$\theta = \pi/2$	$\varphi = 3\pi/2$	$ -i\rangle$
$\hat{z} = (0, 0, 1)$	$\theta = 0$		$ 0\rangle$
$-\hat{z} = (0, 0, -1)$	$\theta = \pi$		$ 1\rangle$

Tabela 3.1: Intersecção da esfera de Bloch com os eixos coordenados.

Observe que os vetores da base \mathcal{X} correspondem às intersecções da esfera com o eixo x . De forma similar, isso vale para as bases \mathcal{Y} e \mathcal{Z} , que correspondem às intersecções da esfera com os eixos y e z , respectivamente. Essas bases foram abordadas na seção 2.2.3.

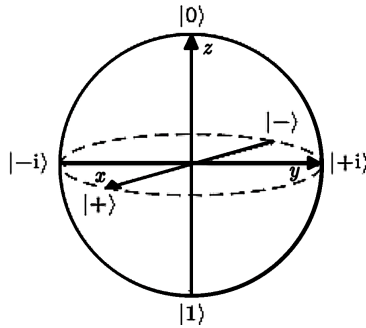


Figura 3.3: Pontos destacados na esfera de Bloch.

Projeções nos eixos coordenados

As projeções nos eixos x , y e z de um ponto \vec{r} na superfície da esfera de Bloch são dadas pelas coordenadas esféricas:

$$\begin{aligned}\vec{r} &= r_x \hat{x} + r_y \hat{y} + r_z \hat{z} \\ &= \sin \theta \cos \varphi \hat{x} + \sin \theta \sin \varphi \hat{y} + \cos \theta \hat{z}, \quad \theta \in [0, \pi], \varphi \in [0, 2\pi) .\end{aligned}$$

Essas projeções correspondem aos valores esperados dos operadores hermitianos X , Y e Z de Pauli:

$$\begin{aligned} r_x &= \sin \theta \cos \varphi = \langle X \rangle \\ r_y &= \sin \theta \sin \varphi = \langle Y \rangle \\ r_z &= \cos \theta = \langle Z \rangle . \end{aligned}$$

3.2.4 Justificativas para a seção anterior

Proposição 3.1. *O estado de um qubit pode ser escrito, a menos de uma fase global, como*

$$|\psi\rangle = \cos(\theta/2) |0\rangle + e^{i\varphi} \sin(\theta/2) |1\rangle , \quad \theta \in [0, \pi], \varphi \in [0, 2\pi) .$$

Demonstração. Seja $|\psi\rangle = a|0\rangle + b|1\rangle$ com $a, b \in \mathbb{C}$ e $|a|^2 + |b|^2 = 1$. Os coeficientes a e b são números complexos, portanto admitem módulo e fase complexa, podendo ser escritos na forma polar como

$$a = \rho_a e^{i\theta_a} , \quad b = \rho_b e^{i\theta_b} ,$$

com $\rho_a, \rho_b \geq 0$. Tem-se

$$|a|^2 + |b|^2 = 1 \implies \rho_a^2 + \rho_b^2 = 1 \implies \begin{cases} \rho_a = \cos(\theta/2) \\ \rho_b = \sin(\theta/2) \end{cases} \quad \theta \in [0, \pi] .$$

Portanto, pode-se escrever

$$\begin{aligned} |\psi\rangle &= \cos(\theta/2) e^{i\theta_a} |0\rangle + \sin(\theta/2) e^{i\theta_b} |1\rangle \\ &= e^{i\theta_a} \left(\cos(\theta/2) |0\rangle + \sin(\theta/2) e^{i(\theta_b - \theta_a)} |1\rangle \right) . \end{aligned}$$

Como a fase global não tem efeito físico, pode-se multiplicar por um fator $e^{-i\theta_a} |0\rangle$ para que o coeficiente de $|0\rangle$ se torne um número real positivo. Definindo $\varphi = \theta_b - \theta_a$, pode-se dizer, sem prejuízo, que $\varphi \in [0, 2\pi)$. Dessa forma, o estado $|\psi\rangle$ fica

$$|\psi\rangle = \cos(\theta/2) |0\rangle + e^{i\varphi} \sin(\theta/2) |1\rangle , \quad \theta \in [0, \pi], \varphi \in [0, 2\pi) . \quad \square$$

Proposição 3.2. *Seja \vec{r} um vetor na superfície da esfera de Bloch correspondendo ao estado $|\psi\rangle$. As coordenadas x , y e z desse vetor correspondem aos valores esperados dos operadores X , Y e Z de Pauli referentes ao estado $|\psi\rangle$:*

$$\begin{aligned} r_x &= \sin \theta \cos \varphi = \langle X \rangle \\ r_y &= \sin \theta \sin \varphi = \langle Y \rangle \\ r_z &= \cos \theta = \langle Z \rangle . \end{aligned}$$

Demonstração. Seja $|\psi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi}\sin(\theta/2)|1\rangle$. Calculando o valor esperado dos operadores X , Y e Z para esse estado, tem-se:

$$\begin{aligned}
 \langle X \rangle &= \langle \psi | X | \psi \rangle \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) X \left(\cos(\theta/2) | 0 \rangle + e^{i\varphi} \sin(\theta/2) | 1 \rangle \right) \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) \left(\cos(\theta/2) | 1 \rangle + e^{i\varphi} \sin(\theta/2) | 0 \rangle \right) \\
 &= 0 + \cos(\theta/2)e^{i\varphi}\sin(\theta/2) + 0 + e^{-i\varphi}\sin(\theta/2)\cos(\theta/2) \\
 &= (e^{i\varphi} + e^{-i\varphi})\sin(\theta/2)\cos(\theta/2) \\
 &= 2\cos\varphi\sin(\theta/2)\cos(\theta/2) \\
 &= \sin\theta\cos\varphi \\
 &= r_x
 \end{aligned}$$

$$\begin{aligned}
 \langle Y \rangle &= \langle \psi | Y | \psi \rangle \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) Y \left(\cos(\theta/2) | 0 \rangle + e^{i\varphi} \sin(\theta/2) | 1 \rangle \right) \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) \left(i\cos(\theta/2) | 1 \rangle - ie^{i\varphi}\sin(\theta/2) | 0 \rangle \right) \\
 &= 0 - i\cos(\theta/2)e^{i\varphi}\sin(\theta/2) + 0 + ie^{-i\varphi}\sin(\theta/2)\cos(\theta/2) \\
 &= -i(e^{i\varphi} - e^{-i\varphi})\sin(\theta/2)\cos(\theta/2) \\
 &= -i(2i\sin\varphi)\sin(\theta/2)\cos(\theta/2) \\
 &= \sin\theta\sin\varphi \\
 &= r_y
 \end{aligned}$$

$$\begin{aligned}
 \langle Z \rangle &= \langle \psi | Z | \psi \rangle \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) Z \left(\cos(\theta/2) | 0 \rangle + e^{i\varphi} \sin(\theta/2) | 1 \rangle \right) \\
 &= \left(\cos(\theta/2) \langle 0 | + e^{-i\varphi} \sin(\theta/2) \langle 1 | \right) \left(\cos(\theta/2) | 0 \rangle - e^{i\varphi} \sin(\theta/2) | 1 \rangle \right) \\
 &= \cos(\theta/2)\cos(\theta/2) + 0 + 0 - e^{-i\varphi}\sin(\theta/2)e^{i\varphi}\sin(\theta/2) \\
 &= \cos^2(\theta/2) - \sin^2(\theta/2) \\
 &= \sin\theta \\
 &= r_z
 \end{aligned}$$

Nas contas, foram usadas as seguintes identidades complexas:

$$\cos\varphi = \frac{e^{i\varphi} + e^{-i\varphi}}{2}, \quad \sin\varphi = \frac{e^{i\varphi} - e^{-i\varphi}}{2i}. \quad \square$$

3.3 Notação de Circuitos

Os detalhes da notação utilizada para representar circuitos quânticos serão vistos nesta seção. A referência utilizada nesta parte é o capítulo 2 do livro [17].

Um circuito quântico é ilustrado na figura 3.4 disposta abaixo. Alguns detalhes pertinentes são abordados na sequência.

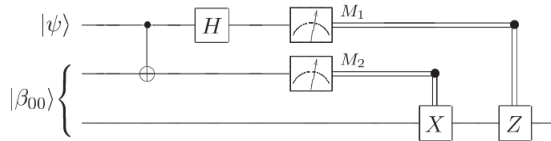


Figura 3.4: Exemplo de circuito quântico. Fonte: [15], p. 27.

Entradas e saídas

O circuito deve conter o mesmo número de entradas e saídas (às vezes podem estar omitidas quando não utilizadas). Cada qubit é representado por uma linha horizontal, e linhas duplas representam bits clássicos. Pode-se pôr rótulos nos qubits para indicar em que estado se encontram na entrada.

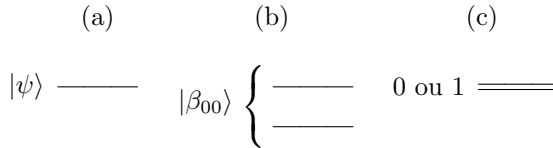


Figura 3.5: (a) Linhas horizontais simples representam 1 qubit. (b) Representação de 2 qubits com o rótulo representando o estado de Bell $|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$; a notação com chave é útil para representar entradas compostas por estados emaranhados. (c) Linhas horizontais duplas representam 1 bit clássico, ou um cbit.

Os qubits de entrada podem se encontrar em estados $|0\rangle$, $|1\rangle$ ou em superposições desses estados. Também podem encontrar-se em estados emaranhados, não possíveis de se exprimir como produto tensorial de estados de 1 qubit.

Sequência de operações

A passagem do tempo, e portanto a sequência de operações, é representada da esquerda para a direita. Ocasionalmente pode-se representar o circuito na forma vertical, e a passagem do tempo é representada de cima para baixo.

Símbolos para Portas Lógicas Quânticas

As portas lógicas quânticas são representadas por caixas contendo o mesmo

número de entradas e saídas. As portas lógicas controladas são portas lógicas de mais de 1 qubit em que pelo menos um dos qubits age como controle; o qubit de controle é representado por um círculo (mais detalhes serão vistos posteriormente).

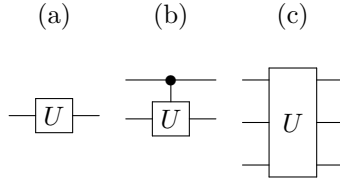


Figura 3.6: Exemplos de portas lógicas quânticas. (a) Porta lógica de 1 qubit. (b) Porta lógica controlada de 2 qubits. (c) Porta lógica de 3 qubits.

Medições de qubits

As medições são as únicas operações potencialmente irreversíveis de um circuito quântico. Em geral são realizadas na base computacional $|0\rangle$ e $|1\rangle$. A notação para medições é ilustrada abaixo, em que, novamente, fica implícito que a base de medidas é a computacional. Após a medição na base computacional, o resultado é um bit clássico.

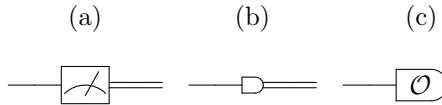


Figura 3.7: Notações possíveis para medição de qubits. Após a medição na base canônica, o resultado é um cbit. (a) e (b) Medição na base canônica. (c) Medição de um observável \mathcal{O} especificado.

3.4 Portas Lógicas Quânticas

As portas lógicas quânticas são operações unitárias aplicadas em um ou mais qubits. Nesta seção, essas portas lógicas são apresentadas em detalhes.

3.4.1 Portas Lógicas de 1 Qubit

Estas portas lógicas atuam em 1 qubit apenas. São descritas por matrizes unitárias 2×2 . As portas lógicas apresentadas nesse tópico são as portas X , Y e Z de Pauli (também denotadas por σ_x , σ_y e σ_z , respectivamente), a porta H de Hadamard, a porta de fase ou porta S e a porta $\frac{\pi}{8}$ ou porta T .

Porta X de Pauli ou NOT Quântica

A porta X de Pauli é a operação unitária de 1 qubit que, na base computacional, é representada pela matriz de Pauli $X = \sigma_x$. Algumas informações dessa porta estão resumidas na figura abaixo.

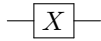

Símbolo	Matriz	Comportamento
	$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$X 0\rangle = 1\rangle$ $X 1\rangle = 0\rangle$
	Notação alternativa	$X b\rangle = \bar{b}\rangle$ $b=0,1 ; \bar{b}=\text{NOT}(b)$
	$X = \sigma_x$	$X +\rangle = +\rangle$ $X -\rangle = - -\rangle$

Figura 3.8: Porta X ou NOT quântica.

Porta Y de Pauli

A porta Y de Pauli é a operação unitária de 1 qubit que, na base computacional, é representada pela matriz de Pauli $Y = \sigma_y$. Algumas informações dessa porta estão dispostas abaixo.

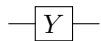
Símbolo	Matriz	Comportamento
	$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	$Y 0\rangle = i 1\rangle$ $Y 1\rangle = -i 0\rangle$
	Notação alternativa	$Y b\rangle = (-1)^b i \bar{b}\rangle$ $b=0,1 ; \bar{b}=\text{NOT}(b)$
	$Y = \sigma_y$	$Y +\rangle = -i -\rangle$ $Y -\rangle = i +\rangle$

Figura 3.9: Porta Y .

Porta Z de Pauli

A porta Z de Pauli é a operação unitária de 1 qubit que é representada na base computacional pela matriz de Pauli $Z = \sigma_z$. Essa porta introduz uma fase relativa de π , o que corresponde a multiplicar o $|1\rangle$ por $-1 = e^{i\pi}$, como se pode observar no quadro a seguir.


Símbolo	Matriz	Comportamento
	$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$Z 0\rangle = 0\rangle$ $Z 1\rangle = - 1\rangle$
	Notação alternativa	$Z b\rangle = (-1)^b b\rangle$ $b=0,1$
	$Z = \sigma_z$	$Z +\rangle = -\rangle$ $Z -\rangle = +\rangle$

Figura 3.10: Porta Z .

Porta Hadamard

A porta de Hadamard é uma operação unitária de 1 qubit representada na base computacional pela matriz de Hadamard H . Essa matriz, definida abaixo, também realiza mudança de base de $\mathcal{I} = \{|0\rangle, |1\rangle\}$ para $\mathcal{X} = \{|+\rangle, |-\rangle\}$ e vice-versa, como visto no exemplo 1.5.


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

Figura 3.11: Porta de Hadamard.

Porta de Fase ou Porta S

A porta S introduz uma fase relativa de $\frac{\pi}{2}$ no qubit em que atua, levando um estado $a|0\rangle + b|1\rangle$ em um estado $a|0\rangle + ib|1\rangle$, já que $i = e^{i\frac{\pi}{2}}$. Os detalhes pertinentes a essa porta lógica estão dispostos abaixo.


Símbolo	Matriz	Comportamento
	$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	$S 0\rangle = 0\rangle$ $S 1\rangle = i 1\rangle$

Figura 3.12: Porta de fase ou porta S .

Porta T ou Porta $\frac{\pi}{8}$

A porta T , também conhecida como porta $\frac{\pi}{8}$, é uma porta lógica que introduz uma fase relativa de $\frac{\pi}{4}$, levando um estado $a|0\rangle + b|1\rangle$ em $a|0\rangle + e^{i\frac{\pi}{4}}b|1\rangle$.

O nome $\frac{\pi}{8}$ dessa porta se deve ao fato de poder ser escrita na forma

$$T = e^{i\frac{\pi}{8}} \begin{bmatrix} e^{-i\frac{\pi}{8}} & 0 \\ 0 & e^{i\frac{\pi}{8}} \end{bmatrix}.$$

Isso significa que, a menos de uma fase global, essa operação realiza uma mudança de fase de $+\frac{\pi}{8}$ no estado $|0\rangle$ e de $-\frac{\pi}{8}$ no estado $|1\rangle$.


Símbolo	Matriz	Comportamento
	$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix}$	$T 0\rangle = 0\rangle$ $T 1\rangle = e^{i\frac{\pi}{4}} 1\rangle$

Figura 3.13: Porta $\frac{\pi}{8}$ ou porta T .

Porta de Fase θ

A porta de fase pode ser generalizada para uma fase arbitrária θ . Nesse caso, a aplicação dessa porta, denotada por $S(\theta)$ leva um estado $a|0\rangle + b|1\rangle$ em $a|0\rangle + e^{i\theta}b|1\rangle$. A matriz que realiza isso é mostrada a seguir.


Símbolo	Matriz	Comportamento
	$S(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix}$	$S 0\rangle = 0\rangle$ $S 1\rangle = e^{i\theta} 1\rangle$

Figura 3.14: Porta de fase $S(\theta)$.

Como casos particulares, tem-se

$$Z = S(\pi), \quad S = S\left(\frac{\pi}{2}\right) \text{ e } T = S\left(\frac{\pi}{4}\right).$$

3.4.2 Portas Lógicas de 2 Qubits

As portas lógicas de 2 qubits são realizadas por matrizes unitárias 4×4 . As principais operações são CNOT (NOT controlada), Z controlada e SWAP, descritas abaixo.

Porta CNOT

A porta CNOT, ou NOT controlada, é uma porta de 2 qubits em que um deles exerce a função de controle e o outro, a de alvo. Em geral, quando não especificado, o primeiro qubit é o controle e o segundo, o alvo. Se o qubit de controle for $|0\rangle$, nada acontece com o qubit alvo. Se o controle for $|1\rangle$, a porta NOT quântica (porta X de Pauli) é aplicada ao alvo:

$$\text{CNOT } |0\rangle_1 |1\rangle_2 = |0\rangle_1 |1\rangle_2, \quad \text{CNOT } |1\rangle_1 |0\rangle_2 = X_2 |1\rangle_1 |0\rangle_2 = |1\rangle_1 |1\rangle_2.$$

Esse comportamento é análogo à entrada “enable” em circuitos digitais clássicos, que permite a ação do circuito se está habilitada em 1, ou nada acontece, se o enable é 0. A novidade na Computação Quântica é que a entrada de controle é um qubit e pode, portanto, se encontrar em uma superposição de estados, como $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$. A aplicação da porta CNOT, em casos como esse, ficaria

$$\begin{aligned} \text{CNOT } \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} |1\rangle_2 &= \frac{1}{\sqrt{2}} (\text{CNOT } |0\rangle_1 |1\rangle_2 + \text{CNOT } |1\rangle_1 |1\rangle_2) \\ &= \frac{1}{\sqrt{2}} (|0\rangle_1 |1\rangle_2 + |1\rangle_1 |0\rangle_2). \end{aligned}$$

A porta CNOT tem suas informações resumidas no quadro abaixo.

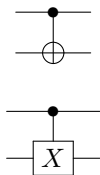
Símbolo	Matriz	Comportamento
	$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$	$\text{CNOT } 00\rangle = 00\rangle$ $\text{CNOT } 01\rangle = 01\rangle$ $\text{CNOT } 10\rangle = 11\rangle$ $\text{CNOT } 11\rangle = 10\rangle$
		$\text{CNOT } 0b\rangle = 0b\rangle$ $\text{CNOT } 1b\rangle = 1\bar{b}\rangle$ $b=0,1; \bar{b}=\text{NOT}(b)$ $\text{CNOT } a,b\rangle = a, a \oplus b\rangle$ $a,b=0,1$ $\oplus = \text{XOR} = \text{adição mod } 2$

Figura 3.15: Porta CNOT.

A porta CNOT também pode aparecer com o controle no segundo qubit e alvo no primeiro qubit. Nesse caso, podem ser usados índices no símbolo

CNOT para especificar o controle e o alvo em situações mais específicas. Por exemplo:

$$\text{CNOT}_{1,2} \quad \begin{array}{c} \bullet \\ | \\ \oplus \end{array}$$

(= CNOT)

$$\text{CNOT}_{2,1} \quad \begin{array}{c} \oplus \\ | \\ \bullet \end{array}$$

Uma notação semelhante pode ser usada em outras portas controladas para especificar o qubit de controle e o de alvo.

Porta Z Controlada

A porta Z controlada também atua em 2 qubits, um deles com função de controle e o outro, de alvo. Se o controle (o primeiro qubit) for $|0\rangle$, o alvo (segundo qubit) não se modifica, e se o controle for $|1\rangle$, aplica-se uma porta Z de Pauli ao alvo, como se pode ver no quadro a seguir. Se os qubits estiverem em superposição, basta usar a linearidade do operador CZ e atuar em cada estado da base computacional isoladamente:

$$\begin{aligned} CZ \frac{|0\rangle_1 + |1\rangle_1}{\sqrt{2}} |1\rangle_2 &= \frac{1}{\sqrt{2}} (CZ |0\rangle_1 |1\rangle_2 + CZ |1\rangle_1 |1\rangle_2) \\ &= \frac{1}{\sqrt{2}} (|0\rangle_1 |1\rangle_2 - |1\rangle_1 |1\rangle_2) . \end{aligned}$$

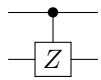
Símbolo	Matriz	Comportamento
	$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{aligned} CZ 00\rangle &= 00\rangle \\ CZ 01\rangle &= 01\rangle \\ CZ 10\rangle &= 10\rangle \\ CZ 11\rangle &= - 11\rangle \\ \\ CZ 0b\rangle &= 0b\rangle \\ CZ 1b\rangle &= Z_2 1b\rangle = 1\rangle (Z b\rangle) \\ &\quad b=0,1 \end{aligned}$

Figura 3.16: Porta Z controlada.

Porta SWAP

A porta SWAP troca o estado de dois qubits, levando $|\phi\rangle |\psi\rangle$ em $|\psi\rangle |\phi\rangle$.

Símbolo

Matriz

Comportamento



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

$$\text{SWAP} |00\rangle = |00\rangle$$

$$\text{SWAP} |01\rangle = |10\rangle$$

$$\text{SWAP} |10\rangle = |01\rangle$$

$$\text{SWAP} |11\rangle = |11\rangle$$

$$\text{SWAP} |ab\rangle = |ba\rangle$$

$a, b=0,1$

Figura 3.17: Porta SWAP.

3.4.3 Portas Lógicas de 3 Qubits

As principais operações em 3 qubits são as portas Toffoli, também conhecida por CCNOT (CNOT controlada), e Fredkin, ou CSWAP (SWAP controlada).

Porta Toffoli ou CCNOT

A porta Toffoli é uma operação linear que envolve 3 qubits. Dois deles funcionam como controle e um, como alvo. O alvo só é modificado (pela aplicação da porta X) se o estado dos dois controles for $|1\rangle|1\rangle$.

Símbolo

Matriz

Comportamento



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

$$\text{CCNOT} |00c\rangle = |00c\rangle$$

$$\text{CCNOT} |01c\rangle = |01c\rangle$$

$$\text{CCNOT} |10c\rangle = |10c\rangle$$

$$\text{CCNOT} |11c\rangle = |11\bar{c}\rangle$$

$$c=0,1 ; \bar{c}=\text{NOT}(c)$$

$$\begin{aligned} \text{CCNOT} |a, b, c\rangle \\ = |a, b, (a \cdot b) \oplus c\rangle \end{aligned}$$

$$a, b, c=0,1$$

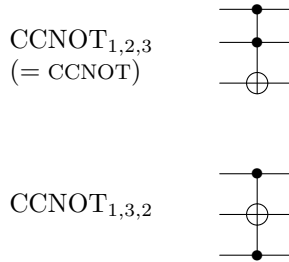
$$\cdot = \text{AND}$$

$$\oplus = \text{XOR} = \text{adição mod } 2$$

Figura 3.18: Porta Toffoli ou CCNOT (CNOT controlada).

Novamente, quando não houver outra indicação, os qubits de controle são o primeiro e o segundo, e o alvo é o terceiro qubit. Pode-se especificar

os qubits de controle e de alvo por meio de índices, como por exemplo:



Porta Fredkin ou CSWAP

A porta de Fredkin possui um qubit de controle e dois alvos. Se o controle for $|1\rangle$, uma porta SWAP atua nos alvos.


Símbolo	Matriz	Comportamento
	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\text{CSWAP } 0bc\rangle = 0bc\rangle$ $\text{CSWAP } 1bc\rangle = 1cb\rangle$ $b, c=0,1$

Figura 3.19: Porta Fredkin ou CSWAP (SWAP controlada).

3.5 Identidades de Circuitos

Esta seção reúne algumas propriedades das portas lógicas quânticas vistas anteriormente. Também são apresentadas identidades úteis para se manipular circuitos quânticos.

Proposição 3.3 (Identidades para as matrizes de Pauli). *Valem as seguintes relações para as matrizes de Pauli X , Y e Z .*

$$\begin{aligned} X^2 &= I & XY &= iZ \\ Y^2 &= I & YZ &= iX \\ Z^2 &= I & ZX &= iY \end{aligned}$$

Demonstração. A prova se dá por cálculo direto.

$$\begin{aligned} XX &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I & XY &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} i & 0 \\ 0 & -i \end{bmatrix} = iZ \\ YY &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I & YZ &= \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} = iX \\ ZZ &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I & ZX &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = iY \end{aligned}$$

□

Proposição 3.4 (Matrizes de Pauli na base $|+\rangle, |-\rangle$). *Fazendo-se a mudança de base de $|0\rangle, |1\rangle$ para $|+\rangle, |-\rangle$ por meio da matriz H , obtém-se para as matrizes de Pauli:*

$$HXH = Z$$

$$HYH = -Y$$

$$HZH = X$$

Portanto, na nova base $|+\rangle, |-\rangle$, a matriz do operador X é Z , a matriz de Y é $-Y$ e a matriz de Z é X .

Demonstração.

$$\begin{aligned} HXH &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = Z \\ HYH &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -i & i \\ i & i \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 0 & 2i \\ -2i & 0 \end{bmatrix} = -\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} = -Y \\ HZH &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = X \end{aligned}$$

□

Proposição 3.5. *A porta de Hadamard é sua própria inversa.*

$$H^2 = I$$

Demonstração. Pode-se verificar fazendo a conta com matrizes diretamente.

$$HH = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = I$$

Outra maneira de se verificar isso é perceber que H é hermitiana ($H^\dagger = H$) e unitária ($H^{-1} = H^\dagger$), de forma que $H^{-1} = H^\dagger = H$.

□

Proposição 3.6 (Relação entre as portas S e T). *Vale que:*

$$T^2 = S$$

Demonstração. $T^2 = \begin{bmatrix} 1 & 0 \\ 1 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & e^{i\frac{\pi}{4}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & e^{i\frac{\pi}{2}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & i \end{bmatrix} = S$ □

Proposição 3.7. *As portas CNOT e SWAP são suas próprias inversas.*

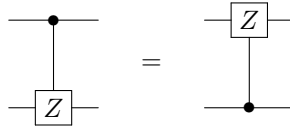
$$\text{CNOT}^2 = I \quad \text{SWAP}^2 = I$$

Demonstração. Ambas operações são hermitianas (em particular, são matrizes de coeficientes reais e simétricas) e unitárias (as colunas são vetores ortonormais), portanto vale o mesmo argumento dado para H :

$$\begin{aligned} \text{CNOT}^\dagger &= \text{CNOT}, \text{CNOT}^{-1} = \text{CNOT}^\dagger & \Rightarrow & \text{CNOT}^{-1} = \text{CNOT}^\dagger = \text{CNOT} \\ \text{SWAP}^\dagger &= \text{SWAP}, \text{SWAP}^{-1} = \text{SWAP}^\dagger & \Rightarrow & \text{SWAP}^{-1} = \text{SWAP}^\dagger = \text{SWAP} \end{aligned}$$

□

Proposição 3.8. *É possível intercambiar alvo e controle na porta Z controlada.*



Demonstração. Para mostrar que dois operadores são iguais, basta verificar que fornecem mesmo resultado nos vetores da base computacional.

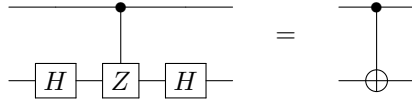
$$\begin{aligned} |00\rangle &\xrightarrow{CZ_{1,2}} |00\rangle \\ |01\rangle &\xrightarrow{CZ_{1,2}} |01\rangle \\ |10\rangle &\xrightarrow{CZ_{1,2}} Z_2 |10\rangle = |10\rangle \\ |11\rangle &\xrightarrow{CZ_{1,2}} Z_2 |11\rangle = -|11\rangle, \end{aligned}$$

$$\begin{aligned} |00\rangle &\xrightarrow{CZ_{2,1}} |00\rangle \\ |01\rangle &\xrightarrow{CZ_{2,1}} Z_1 |01\rangle = |01\rangle \\ |10\rangle &\xrightarrow{CZ_{2,1}} |10\rangle \\ |11\rangle &\xrightarrow{CZ_{2,1}} Z_1 |11\rangle = -|11\rangle. \end{aligned}$$

Isso implica igualdade entre os operadores $CZ_{1,2}$ e $CZ_{2,1}$.

□

Proposição 3.9. *A porta CNOT pode ser obtida usando-se uma porta Z controlada:*



Demonstração. Os dois circuitos representam operadores lineares em 2 qubits. É suficiente, pois, verificar que o comportamento dos dois circuitos é o mesmo na base computacional, pois as transformações lineares são especificadas de forma única por seu resultado em uma base qualquer (ver seção 1.3.4). Nesta demonstração, também serão usadas as propriedades $HH = I$ e $HZH = X$, verificadas nas proposições 3.4 e 3.5.

De fato, analisando o efeito do circuito para os vetores da base computacional, obtém-se o seguinte. Seja $b = 0, 1$. Se o primeiro qubit for $|0\rangle$, a porta Z controlada não surtirá efeito no segundo qubit, de forma que

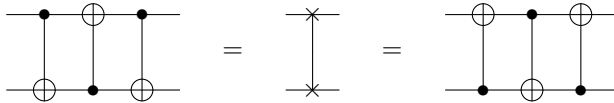
$$|\psi_f\rangle_{1,2} = H_2 H_2 |0b\rangle = I_2 |0b\rangle = |0b\rangle .$$

Se o primeiro qubit for $|1\rangle$, a porta Z controlada atuará, resultando em

$$|\psi_f\rangle_{1,2} = H_2 Z_2 H_2 |1b\rangle = X_2 |1b\rangle = |1\bar{b}\rangle .$$

O efeito final para esses vetores da base é o mesmo que o de uma porta CNOT, dessa forma, os dois circuitos do enunciado são equivalentes. \square

Proposição 3.10. *A porta SWAP pode ser construída por 3 portas CNOT:*



Demonstração. Verificando a primeira igualdade nos vetores da base computacional:

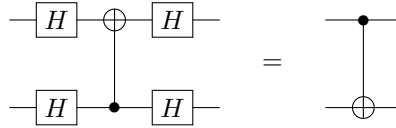
$ 00\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 00\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 00\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 00\rangle$
$ 01\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 01\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 11\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 10\rangle$
$ 10\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 11\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 01\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 01\rangle$
$ 11\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 10\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 10\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 11\rangle$

Verificando a segunda igualdade nos vetores da base computacional:

$ 00\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 00\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 00\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 00\rangle$
$ 01\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 11\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 10\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 10\rangle$
$ 10\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 10\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 11\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 01\rangle$
$ 11\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 01\rangle$	$\xrightarrow{\text{CNOT}_{1,2}}$	$ 01\rangle$	$\xrightarrow{\text{CNOT}_{2,1}}$	$ 11\rangle$

Percebe-se, então, que o efeito dessas sequências de portas CNOT é o mesmo de uma porta SWAP. \square

Proposição 3.11. *Dependendo da base considerada, os papeis de controle e alvo da porta CNOT se invertem. Mais precisamente, vale a seguinte igualdade de circuitos:*



Demonstração. Primeiramente, perceba que o efeito da porta Hadamard é de mudança de base saindo de $\mathcal{I} = \{|0\rangle, |1\rangle\}$ para $\mathcal{X} = \{|+\rangle, |-\rangle\}$ e vice versa, portanto para 2 qubits vale:

$$\begin{array}{llll}
 |00\rangle & \xrightarrow{H_1 H_2} & |++\rangle & \xrightarrow{H_1 H_2} & |00\rangle \\
 |01\rangle & \xrightarrow{H_1 H_2} & |+-\rangle & \xrightarrow{H_1 H_2} & |01\rangle \\
 |10\rangle & \xrightarrow{H_1 H_2} & |-+\rangle & \xrightarrow{H_1 H_2} & |10\rangle \\
 |11\rangle & \xrightarrow{H_1 H_2} & |--\rangle & \xrightarrow{H_1 H_2} & |11\rangle .
 \end{array} \quad (*)$$

Aplicando-se $\text{CNOT}_{2,1}$ à base $\mathcal{X} \otimes \mathcal{X} = \{|++\rangle, |+-\rangle, |-+\rangle, |--\rangle\}$, obtém-se que:

$$\begin{aligned}
 \text{CNOT}_{2,1} |++\rangle &= \text{CNOT}_{2,1} \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\
 &= \frac{1}{2} (\text{CNOT}_{2,1} |00\rangle + \text{CNOT}_{2,1} |01\rangle + \text{CNOT}_{2,1} |10\rangle + \text{CNOT}_{2,1} |11\rangle) \\
 &= \frac{1}{2} (|00\rangle + |11\rangle + |10\rangle + |01\rangle) \\
 &= \frac{1}{2} (|0\rangle (|0\rangle + |1\rangle) + |1\rangle (|0\rangle + |1\rangle)) \\
 &= \frac{1}{2} (|0\rangle + |1\rangle)(|0\rangle + |1\rangle) = |++\rangle ,
 \end{aligned}$$

$$\begin{aligned}
 \text{CNOT}_{2,1} |+-\rangle &= \text{CNOT}_{2,1} \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \\
 &= \frac{1}{2} (\text{CNOT}_{2,1} |00\rangle - \text{CNOT}_{2,1} |01\rangle + \text{CNOT}_{2,1} |10\rangle - \text{CNOT}_{2,1} |11\rangle) \\
 &= \frac{1}{2} (|00\rangle - |11\rangle + |10\rangle - |01\rangle) \\
 &= \frac{1}{2} (|0\rangle (|0\rangle - |1\rangle) + |1\rangle (|0\rangle - |1\rangle)) \\
 &= \frac{1}{2} (|0\rangle + |1\rangle)(|0\rangle - |1\rangle) = |+-\rangle ,
 \end{aligned}$$

$$\begin{aligned}
\text{CNOT}_{2,1} | - + \rangle &= \text{CNOT}_{2,1} \frac{1}{2} (|00\rangle + |01\rangle - |10\rangle - |11\rangle) \\
&= \frac{1}{2} (\text{CNOT}_{2,1} |00\rangle + \text{CNOT}_{2,1} |01\rangle - \text{CNOT}_{2,1} |10\rangle - \text{CNOT}_{2,1} |11\rangle) \\
&= \frac{1}{2} (|00\rangle + |11\rangle - |10\rangle - |01\rangle) \\
&= \frac{1}{2} (|0\rangle (|0\rangle - |1\rangle) - |1\rangle (|0\rangle - |1\rangle)) \\
&= \frac{1}{2} (|0\rangle - |1\rangle)(|0\rangle - |1\rangle) = | - - \rangle ,
\end{aligned}$$

$$\begin{aligned}
\text{CNOT}_{2,1} | - - \rangle &= \text{CNOT}_{2,1} \frac{1}{2} (|00\rangle - |01\rangle - |10\rangle + |11\rangle) \\
&= \frac{1}{2} (\text{CNOT}_{2,1} |00\rangle - \text{CNOT}_{2,1} |01\rangle - \text{CNOT}_{2,1} |10\rangle + \text{CNOT}_{2,1} |11\rangle) \\
&= \frac{1}{2} (|00\rangle - |11\rangle - |10\rangle + |01\rangle) \\
&= \frac{1}{2} (|0\rangle (|0\rangle + |1\rangle) - |1\rangle (|0\rangle + |1\rangle)) \\
&= \frac{1}{2} (|0\rangle - |1\rangle)(|0\rangle + |1\rangle) = | - + \rangle .
\end{aligned}$$

Pode-se resumir o resultado dessas contas na seguinte expressão:

$$\begin{array}{ccc}
|+b\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |+b\rangle \\
|-b\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |+\bar{b}\rangle
\end{array} \quad \begin{array}{l} b \in \{+, -\} \\ \bar{+} := - \\ \bar{-} := + \end{array} \quad (**)$$

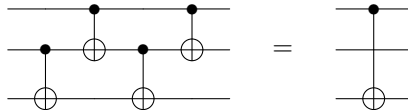
Isto é, a porta $\text{CNOT}_{2,1}$ atuando na base $\mathcal{X} \otimes \mathcal{X}$ tem o mesmo efeito da porta $\text{CNOT}_{1,2}$ atuando na base $\mathcal{I} \otimes \mathcal{I}$, substituindo-se 0 por + e 1 por -.

Portanto, usando-se (*) e (**), o circuito da esquerda do enunciado da proposição fornece

$$\begin{array}{llll}
|00\rangle & \xrightarrow{H_1 H_2} & |++\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |++\rangle & \xrightarrow{H_1 H_2} & |00\rangle \\
|01\rangle & \xrightarrow{H_1 H_2} & |+-\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |+-\rangle & \xrightarrow{H_1 H_2} & |01\rangle \\
|10\rangle & \xrightarrow{H_1 H_2} & |-+\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |--\rangle & \xrightarrow{H_1 H_2} & |11\rangle \\
|11\rangle & \xrightarrow{H_1 H_2} & |--\rangle & \xrightarrow{\text{CNOT}_{2,1}} & |--\rangle & \xrightarrow{H_1 H_2} & |10\rangle ,
\end{array}$$

resultando no mesmo efeito de uma $\text{CNOT}_{1,2}$. □

Proposição 3.12. *É possível implementar uma CNOT com alvo e controle distantes com CNOTs entre qubits adjacentes.*



Demonstração. Sejam $a, b, c = 0, 1$. Aplicando a sequência de CNOTs adjacentes mostradas no enunciado da proposição, obtém-se:

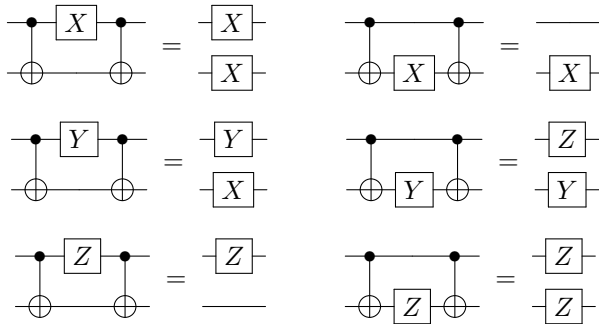
$$\begin{aligned}
 |a, b, c\rangle &\xrightarrow{\text{CNOT}_{2,3}} |a, b, b \oplus c\rangle \\
 &\xrightarrow{\text{CNOT}_{1,2}} |a, a \oplus b, b \oplus c\rangle \\
 &\xrightarrow{\text{CNOT}_{2,3}} |a, a \oplus b, a \oplus b \oplus b \oplus c\rangle \\
 &= |a, a \oplus b, a \oplus c\rangle \\
 &\xrightarrow{\text{CNOT}_{1,2}} |a, a \oplus a \oplus b, a \oplus c\rangle \\
 &= |a, a \oplus a \oplus b, a \oplus c\rangle = \text{CNOT}_{1,3} |a, b, c\rangle .
 \end{aligned}$$

Isso prova a igualdade de circuitos apresentada no enunciado. \square

Proposição 3.13 (Identidades envolvendo CNOT e matrizes de Pauli). *Considere a porta CNOT com controle no qubit 1 e alvo no qubit 2. Valem as seguintes identidades de circuitos:*

$$\begin{aligned}
 \text{CNOT } X_1 \text{ CNOT} &= X_1 X_2 \\
 \text{CNOT } Y_1 \text{ CNOT} &= Y_1 X_2 \\
 \text{CNOT } Z_1 \text{ CNOT} &= Z_1 \\
 \text{CNOT } X_2 \text{ CNOT} &= X_2 \\
 \text{CNOT } Y_2 \text{ CNOT} &= Z_1 Y_2 \\
 \text{CNOT } Z_2 \text{ CNOT} &= Z_1 Z_2 .
 \end{aligned}$$

Graficamente, as igualdades ficam:



Demonstração. A prova dessas igualdades é essencialmente verificar que os circuitos têm o mesmo efeito na base computacional.

$$\text{CNOT } X_1 \text{ CNOT} = X_1 X_2$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{X_1} & |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle = X_1 X_2 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{X_1} & |0\bar{b}\rangle & \xrightarrow{\text{CNOT}} & |0\bar{b}\rangle = X_1 X_2 |1b\rangle \end{array}$$

$$\text{CNOT } X_2 \text{ CNOT} = X_2$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{X_2} & |0\bar{b}\rangle & \xrightarrow{\text{CNOT}} & |0\bar{b}\rangle = X_2 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{X_2} & |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle = X_2 |1b\rangle \end{array}$$

$$\text{CNOT } Y_1 \text{ CNOT} = Y_1 X_2$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{Y_1} & i |1b\rangle & \xrightarrow{\text{CNOT}} & i |1\bar{b}\rangle = Y_1 X_2 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{Y_1} & -i |0\bar{b}\rangle & \xrightarrow{\text{CNOT}} & -i |0\bar{b}\rangle = Y_1 X_2 |1b\rangle \end{array}$$

$$\text{CNOT } Y_2 \text{ CNOT} = Z_1 Y_2$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{Y_2} & (-1)^b i |0\bar{b}\rangle & \xrightarrow{\text{CNOT}} & (-1)^b i |0\bar{b}\rangle = Z_1 Y_2 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{Y_2} & -(-1)^b i |1b\rangle & \xrightarrow{\text{CNOT}} & -(-1)^b i |1\bar{b}\rangle = Z_1 Y_2 |1b\rangle \end{array}$$

Observação: pode-se escrever $Y |b\rangle = (-1)^b i |\bar{b}\rangle$ e $Y |\bar{b}\rangle = -(-1)^b i |b\rangle$.

$$\text{CNOT } Z_1 \text{ CNOT} = Z_1$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{Z_1} & |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle = Z_1 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{Z_1} & -|1\bar{b}\rangle & \xrightarrow{\text{CNOT}} & -|1b\rangle = Z_1 |1b\rangle \end{array}$$

$$\text{CNOT } Z_2 \text{ CNOT} = Z_1 Z_2$$

$$\begin{array}{lclclcl} |0b\rangle & \xrightarrow{\text{CNOT}} & |0b\rangle & \xrightarrow{Z_2} & (-1)^b |0b\rangle & \xrightarrow{\text{CNOT}} & (-1)^b |0b\rangle = Z_1 Z_2 |0b\rangle \\ |1b\rangle & \xrightarrow{\text{CNOT}} & |1\bar{b}\rangle & \xrightarrow{Z_2} & -(-1)^b |1\bar{b}\rangle & \xrightarrow{\text{CNOT}} & -(-1)^b |1b\rangle = Z_1 Z_2 |1b\rangle \end{array}$$

Observação: pode-se escrever $Z |b\rangle = (-1)^b |b\rangle$ e $Z |\bar{b}\rangle = -(-1)^b |\bar{b}\rangle$.

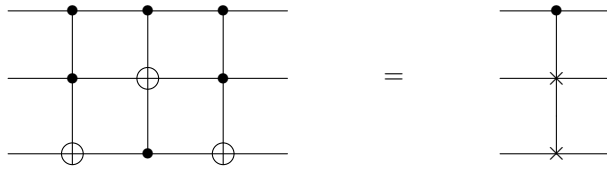
Dessa forma, os circuitos do enunciado têm o mesmo resultado na base computacional, e, portanto, são equivalentes. \square

Proposição 3.14. *As portas Toffoli e Fredkin são suas próprias inversas.*

$$\text{CCNOT}^2 = I \quad \text{CSWAP}^2 = I$$

Demonstração. Análoga à prova da proposição 3.7. \square

Proposição 3.15. *A porta Fredkin pode ser obtida com 3 portas Toffoli:*



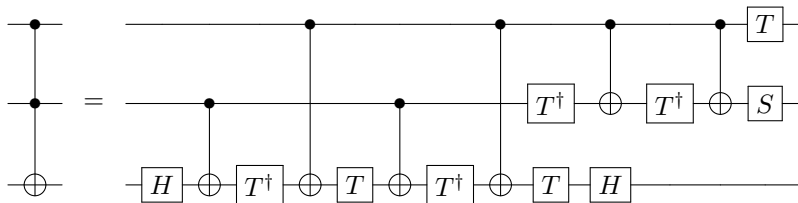
Demonstração. Basta verificar a igualdade para os vetores da base computacional. Sejam $b, c = 0, 1$.

Para os vetores da forma $|0bc\rangle$, a sequência de portas Toffoli não afeta o vetor. O resultado continua sendo $|0bc\rangle$ pois só há modificação quando o AND dos controles for 1 (o que não ocorre em nenhuma das portas Toffoli nesse caso).

Já para os vetores da forma $|1bc\rangle$, as portas Toffoli se ativam dependendo do valor de b e c . O comportamento das portas Toffoli fica semelhante ao das portas CNOT da proposição 3.10 nesse caso em que o primeiro qubit se encontra no estado $|1\rangle$. E esse comportamento, pela referida proposição, é o mesmo de uma porta SWAP.

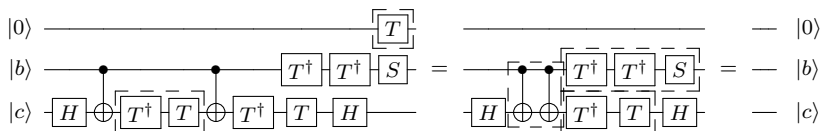
É possível perceber, portanto, que o conjunto de portas Toffoli no enunciado funcionam como um SWAP controlado (isto é, uma porta Fredkin): quando o primeiro qubit é $|0\rangle$, nada acontece, e quando o primeiro qubit é $|1\rangle$, os dois últimos sofrem um efeito igual à aplicação de uma porta SWAP. Logo, verifica-se a equivalência dos circuitos no enunciado da proposição. \square

Proposição 3.16 (Construção da porta Toffoli). *A porta Toffoli pode ser obtida pela seguinte combinação de portas lógicas de 1 e 2 qubits:*



Demonstração. Pode-se verificar essa construção avaliando o efeito nos elementos da base computacional. Sejam $b, c = 0, 1$.

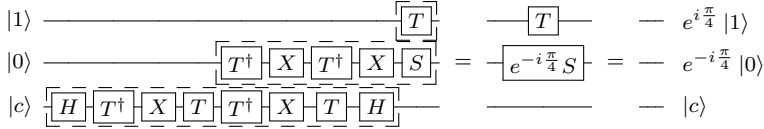
Caso o primeiro qubit seja $|0\rangle$, o circuito simplifica-se para:



Algumas simplificações estão destacadas nas figuras e correspondem a:

$$T|0\rangle = |0\rangle, \quad \text{CNOT}^2 = I, \quad TT^\dagger = I, \quad T^\dagger T^\dagger S = (T^\dagger)^2 T^2 = I, \quad HH = I.$$

Agora será checado o caso em que os dois primeiros qubits são $|10\rangle$. Nesse caso, o circuito original reduz-se a:



As simplificações utilizadas foram:

$$T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle$$

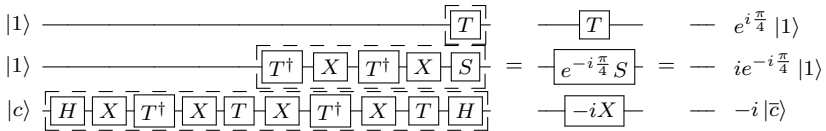
$$\begin{aligned} SXT^\dagger XT^\dagger &= \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 0 & e^{-i\frac{\pi}{4}} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & e^{-i\frac{\pi}{4}} \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} = e^{-i\frac{\pi}{4}} S \end{aligned}$$

$$HTXT^\dagger TXT^\dagger H = HTXXT^\dagger H = HTT^\dagger H = HH = I.$$

O estado ao final do circuito é

$$(e^{i\frac{\pi}{4}}|1\rangle) \otimes (e^{-i\frac{\pi}{4}}|0\rangle) \otimes |c\rangle = |10c\rangle.$$

Resta verificar o caso em que os dois primeiros qubits são $|11\rangle$. Nesse caso, o circuito se reduz a:



Novamente, as simplificações realizadas são:

$$T|1\rangle = e^{i\frac{\pi}{4}}|1\rangle$$

$$SXT^\dagger XT^\dagger|1\rangle = e^{-i\frac{\pi}{4}}S|1\rangle = ie^{-i\frac{\pi}{4}}|1\rangle$$

$$\begin{aligned} TXT^\dagger X &= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} H(TXT^\dagger X)(TXT^\dagger X)H &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} e^{-i\frac{\pi}{4}} & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} -i & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \\ &= H(-iZ)H = -iHZH = -iX, \end{aligned}$$

e o estado final é

$$(e^{i\frac{\pi}{4}} |1\rangle) \otimes (ie^{-i\frac{\pi}{4}} |1\rangle) \otimes (-i |\bar{c}\rangle) = |11\bar{c}\rangle .$$

Termina, portanto, a verificação de que o circuito do enunciado realiza uma porta Toffoli. \square

3.6 Universalidade das Portas Lógicas Quânticas

É útil saber quais conjuntos de portas lógicas quânticas permitem reproduzir qualquer operação unitária U em n qubits. Essa questão é conhecida como *universalidade* de um conjunto de portas lógicas quânticas. A universalidade pode ter um sentido estrito – isto é, uma operação U qualquer pode ser implementada exatamente com um número finito portas lógicas – ou um sentido amplo – isto é, a operação U pode ser aproximada, permitindo-se um erro ε arbitrado, por uma sequência finita de portas lógicas (cujo número é função do erro ε máximo estipulado).

3.6.1 Universalidade de Portas Lógicas na Computação Clássica Reversível

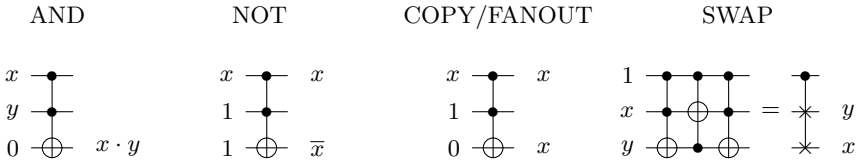
A universalidade na Computação Clássica Não-Reversível está essencialmente decidida nos teoremas A.4 e A.6 da seção A.3.4. Esses teoremas dizem que qualquer função booleana $f: \{0, 1\}^m \rightarrow \{0, 1\}^n$ pode ser realizada por portas NOT e AND, por portas NOT e OR ou apenas por portas NAND (e em todos os casos, acrescenta-se implicitamente as portas SWAP e FANOUT/COPY).

Na Computação Clássica Reversível, as funções booleanas têm o mesmo número de bits na entrada e saída, e as portas lógicas são reversíveis (isto é, conhecendo-se a saída, é possível determinar a entrada que a originou). As portas lógicas NOT, CNOT, SWAP, Toffoli e Fredkin, quando trabalhando apenas com bits e não qubits, são exemplos de portas lógicas clássicas reversíveis.

Teorema 3.17. *A porta Toffoli é universal para a Computação Clássica.*

Demonstração. É possível realizar as portas NOT e AND utilizando a porta Toffoli e acrescentando alguns bits de trabalho (bits com valor fixado

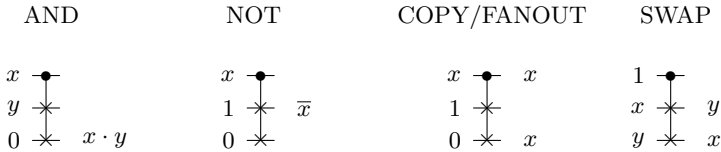
em 0 ou 1 dependendo da necessidade).



Com isso, um circuito booleano qualquer pode ser realizado apenas com portas Toffoli ignorando-se bits extra (“lixo”). \square

Teorema 3.18. *A porta Fredkin é universal para a Computação Clássica.*

Demonstração. Da mesma forma como para a porta Toffoli, é possível realizar as portas NOT e AND utilizando a porta Fredkin e acrescentando alguns bits de trabalho.



Com isso, um circuito booleano qualquer pode ser realizado apenas com portas Fredkin ignorando-se bits extra (“lixo”). \square

Observação 3.19. O preço a se pagar, na Computação Clássica, para que a computação seja reversível é o uso de bits de trabalho (*ancilla bits*) e produção de bits “lixo” (*garbage bits*). Em [18] (capítulo¹ 6, p.12-15) verifica-se a universalidade do conjunto de portas reversíveis Toffoli e NOT para a Computação Clássica Reversível usando-se apenas 1 bit de trabalho.

Observação 3.20. Na referência [18] (capítulo¹ 6, p.11-12) encontra-se uma demonstração de que as portas clássicas reversíveis de 1 e 2 bits não podem formar um conjunto universal.

3.6.2 Universalidade de Portas Lógicas na Computação Quântica

Como as portas Toffoli e Fredkin da Computação Clássica têm análogos quânticos, e tendo em vista a universalidade dessas portas na Computação Clássica, tem-se que a Computação Quântica engloba a Computação Clássica quando lança-se mão de qubits de trabalho e ignoram-se os qubits “lixo” ([3], p.122).

¹Disponível no link:

<http://www.theory.caltech.edu/people/preskill/ph229/notes/chap6.pdf>

É possível verificar que, para a Computação Quântica, todas as portas lógicas de 1 qubit e a porta CNOT formam um conjunto universal no sentido estrito (ou seja, capaz de produzir exatamente, em princípio, qualquer transformação unitária) ([3], p.119-124), o que já não ocorre com a Computação Clássica Reversível (conforme final da seção 3.6.1, observação 3.20).

O conjunto finito de operações X , Y , Z , H , T , S , e CNOT também é universal, agora no sentido amplo (isto é, capaz apenas de aproximar uma operação unitária geral dentro de uma faixa de erro pré-determinada). Esse conjunto não é mínimo, pois algumas dessas portas de 1 qubit podem ser obtidas das demais. Por exemplo: $S = T^2$, $Z = HXH = S^2$, $Y = -iZX$.

As portas lógicas quânticas Toffoli e Hadamard formam um conjunto universal capaz de aproximar qualquer matriz unitária com coeficientes reais. Com o uso de um qubit extra, de trabalho, é possível realizar operações unitárias gerais em função de matrizes unitárias com coeficientes reais (conforme artigo [2], p.2-3).

O assunto de universalidade de portas lógicas quânticas está em desenvolvimento. Não se conhece um algoritmo ou método eficiente capaz de decompor uma matriz unitária U em fatores de um dado conjunto universal. A maioria dos conjuntos finitos que foram provados universais têm suas demonstrações baseadas em argumentos de existência: mostra-se que existe uma sequência de portas que aproxima dentro de um erro dado a porta U , mas não se sabe como encontrar essa aproximação. Para a universalidade de todas as portas de 1 qubit mais a CNOT, a prova é construtiva ([3], p.119-124), mas recorre à disponibilidade de qualquer porta de 1 qubit, além de não ser eficiente em geral (isto é, requerer um número de portas lógicas exponencial no número de qubits n).

3.7 Teorema da Não-Clonagem

Um bit clássico pode ser copiado para servir como entrada em diversas partes de um circuito digital clássico.

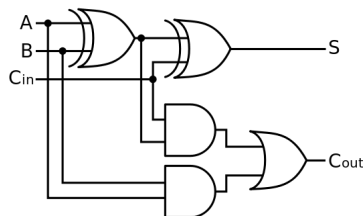


Figura 3.20: Exemplo de como um bit clássico pode ser usado em mais de uma porta lógica. As ramificações, simbolizadas por •, significam criação de cópias do bit.

Pode-se pensar nesse comportamento em termos da porta lógica clássica COPY que devolve à saída duas (ou mais) cópias do bit de entrada.

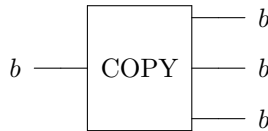


Figura 3.21: Exemplo de função booleana $f(b) = (b, b, b)$, que transforma 1 bit de entrada b em 3 cópias de b à saída.

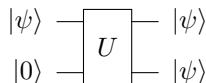
Como na Computação Quântica, o análogo das portas lógicas seriam as operações unitárias sobre qubits, poderia-se cogitar a existência de uma porta lógica quântica de 2 qubits que tivesse como entrada um qubit num estado $|\psi\rangle$ qualquer, a ser copiado, (e outra entrada $|0\rangle$ para completar 2 entradas) e devolvesse 2 qubits no estado $|\psi\rangle$, como ilustrado na figura a seguir.



Figura 3.22: Proposta de porta lógica quântica que copiaria o estado do qubit $|\psi\rangle$ e devolveria 2 qubits no mesmo estado.

No entanto, o chamado *Teorema da Não Clonagem* informa que não existe uma operação unitária capaz de efetuar essa operação para qualquer estado $|\psi\rangle$ de entrada. Desse modo, a cópia de bits não possui análogo na Computação Quântica.

Teorema 3.21 (Teorema da Não Clonagem). *Não existe uma operação unitária que permita copiar o estado de 1 qubit em 2 (ou mais) qubits. Isto é, não existe operação unitária U que satisfaça, para todo estado $|\psi\rangle$ de 1 qubit, o seguinte:*



Demonstração. A prova se dá por redução ao absurdo. Seja U uma tal operação unitária, satisfazendo $U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$ para todo estado $|\psi\rangle$ com $\| |\psi\rangle \| = 1$.

Considere os estados de 1 qubit $|1\rangle$ e $|+\rangle$. Tem-se que

$$\begin{aligned} \langle 10|+0\rangle_{1,2} &= \langle 1|+\rangle_1 \langle 0|0\rangle_2 \\ &= \langle 1|+\rangle \cdot 1 \\ &= \langle 1|\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) \\ &= \frac{1}{\sqrt{2}}. \end{aligned} \quad (*)$$

Por outro lado, como U é unitária, também vale que

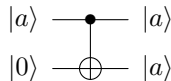
$$\begin{aligned} \langle 10|+0\rangle_{1,2} &= \langle 10|_{1,2} U^\dagger U | +0\rangle_{1,2} \\ &= \langle 11|_{1,2} |++\rangle_{1,2} \\ &= \langle 1|+\rangle \langle 1|+\rangle \\ &= \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}. \end{aligned} \quad (**)$$

Comparando (*) e (**), tem-se que $\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}}$, o que não pode ocorrer. Portanto a hipótese de que existe U como descrito acima é falsa. \square

É possível generalizar essa demonstração para mostrar que não existe operação unitária U tal que $U|\psi\rangle|s\rangle = |\psi\rangle|\psi\rangle$ para todo estado $|\psi\rangle$, e com $|s\rangle$ um estado fixo qualquer ([15], p.532).

Apesar de não ser possível clonar um estado arbitrário, é possível copiar estados na base computacional. De fato, uma porta CNOT é suficiente para realizar isso.

Proposição 3.22 (Clonagem de estados $|0\rangle, |1\rangle$). *O circuito abaixo realiza a clonagem de estados da base computacional. Se $a = 0, 1$, tem-se:*



Demonstração. Segue do comportamento da porta CNOT nos vetores da base computacional:

$$|a\rangle|0\rangle \xrightarrow{\text{CNOT}} |a\rangle|0 \oplus a\rangle = |a\rangle|a\rangle. \quad \square$$

Capítulo 4

Panorama Atual da Computação Quântica

A Computação Quântica é uma tecnologia ainda em estágio inicial, e tem se mostrado uma área de pesquisa estratégica no cenário internacional. Neste capítulo, faz-se um breve estudo das expectativas de mercado em relação à Computação Quântica, em grande parte baseado na Gartner, uma consultoria em Tecnologia da Informação [4]. Apresenta-se também algumas das principais empresas com pesquisas em Computação Quântica, e mostra-se um resumo dos desenvolvimentos realizados nessas empresas tanto em hardware como em software.

4.1 Expectativas de Mercado

A Gartner [4] estima que a Computação Quântica será uma realidade no mercado dentro de 5 a 10 anos. Espera-se que essa tecnologia se torne mais rápida e escalável, de forma a tratar problemas reais que a computação atual não conseguiria abordar satisfatoriamente. A curva da figura 4.1, chamada *hype-cycle*, mostra as tecnologias em alta no mercado e as que já atingiram maturidade. A posição da Computação Quântica é de subida na curva de expectativas.

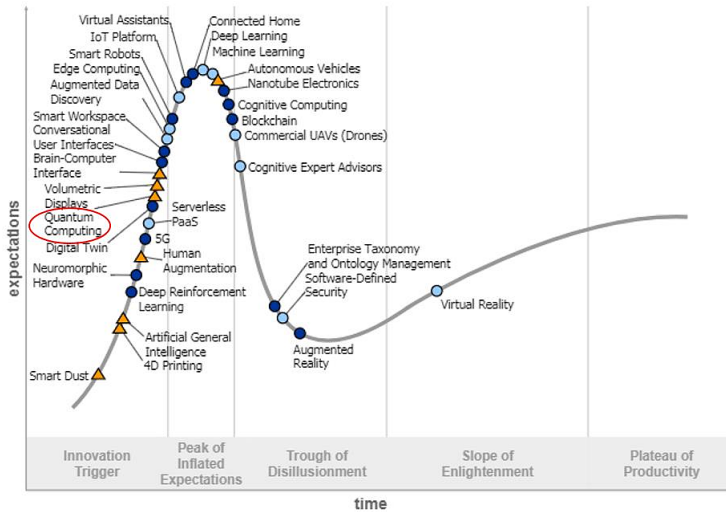


Figura 4.1: *Hype-cycle* para tecnologias atuais. Posição da Computação Quântica destacada. Compare com *Machine Learning* (topo da curva) e *Realidade Virtual* (estabilização/maturidade). Fonte: Gartner [4].

Há uma percepção no mercado que a Computação Quântica gere uma disrupção algo similar às ocorridas na Revoluções Industrial, Tecnológica e Digital. O panorama atual da Computação Quântica é comparável, nessa percepção, ao computador ENIAC da década de 1950 (para comparar, ver figura 4.2). À época, não havia como prever o desenvolvimento tecnológico subsequente, e a capacidade de processamento que se conseguiria obter nas décadas seguintes.

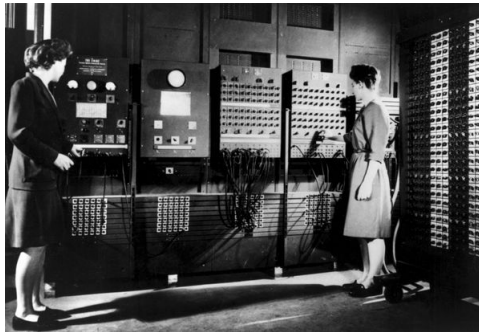


Figura 4.2: ENIAC. Fonte: Computer History Museum [14].

A expectativa é que um modelo híbrido entre Computação Clássica e Quântica seja a tecnologia emergente no cenário atual. A computação seria realizada em um processador quântico para problemas nos quais há vantagens no uso da Computação Quântica, como espera-se que ocorra

com os problemas chamados NP-difíceis.

Dentre as principais aplicações previstas para a Computação Quântica, destaca-se resolução de problemas de otimização, machine learning, desenvolvimento de novos materiais, fármacos e processos químicos. Outra aplicação de destaque é em criptografia. A figura 4.3 mostra aplicações em que se espera utilizar processamento quântico.

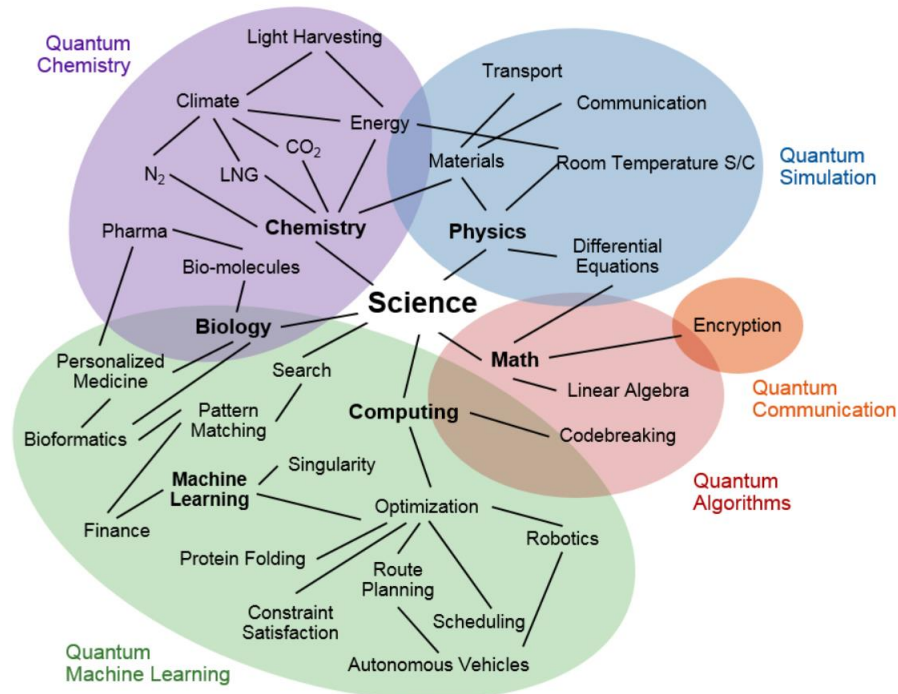


Figura 4.3: Expectativa de aplicação da Computação Quântica. Fonte: Gartner [4], adaptado de Pete Shadbolt e Jeremy O'Brien.

O impacto da Computação Quântica em sistemas de segurança é considerado certo dentro de poucos anos. Algumas tecnologias atuais como a *criptografia RSA* e o *blockchain* (do qual o *bitcoin* faz uso, em particular) seriam vulneráveis à Computação Quântica. Há demanda por protocolos de criptografia que levem em conta a existência da Computação Quântica, isto é, por criptografia *pós-quântica* (*post-quantum/quantum-safe/quantum-proof cryptography*).

4.2 Empresas e Desenvolvimentos Atuais

A Computação Quântica encontra-se em um estágio de desenvolvimento acelerado. Empresas como IBM, Google, Intel e Microsoft, e Startups como Rigetti, têm pesquisas para desenvolvimento de processadores quânticos. Algumas dessas empresas, como a IBM e Rigetti, disponibilizam protótipos de computadores quânticos para acesso pela nuvem ao público. Também há desenvolvimento de simuladores, linguagens de programação para Computação Quântica e kits de softwares por essas empresas. Alguns comentários são desenvolvidos no que segue.

IBM

A IBM utiliza tecnologia de qubits supercondutores, compostos por junções de Josephson. Em 10 de novembro de 2017, foram anunciados protótipos de 50 e de 20 qubits, em uma iniciativa de tornar a computação quântica comercialmente disponível em um futuro próximo.

No projeto IBM Quantum Experience, um computador quântico de 5 qubits está disponível para uso através da nuvem. A programação é feita por uma interface gráfica, que permite realizar simulação e/ou enviar o algoritmo para execução no computador quântico. A programação pode ser realizada em modo texto, com a linguagem OpenQASM (Open Quantum Assembly Language). Em relação à simulação, a IBM também colabora no projeto open source QISKit (Quantum Information Software Kit), uma biblioteca python para Computação e Informação Quântica que funciona em conjunto com OpenQASM.

Fonte:

<http://newsroom.ibm.com/IBM-research?item=30270>

<http://newsroom.ibm.com/>

<https://www.research.ibm.com/ibm-q/>

<https://quantumexperience.ng.bluemix.net/qx/experience>

<https://qiskit.org/>

<https://github.com/QISKit>

<https://github.com/QISKit/openqasm>

Google

O processador quântico Bristlecone, de 72 qubits, foi apresentado em 5 de março de 2018. A tecnologia de qubits da Google é baseada em filmes supercondutores de alumínio em substrato de safira. Para simulação de computadores quânticos, a empresa tem um projeto open source chamado

Quantum Playground, em que é possível simular um computador quântico de até 22 qubits.

Fonte:

<https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>

<https://ai.googleblog.com/2015/03/a-step-closer-to-quantum-computation.html>

<https://www.nature.com/articles/nature14270>

<http://www.quantumplayground.net>

<https://opensource.google.com/projects/quantum-computing-playground>

<https://github.com/gwroblew/Quantum-Computing-Playground>

Intel

Em 8 de janeiro de 2018 foi anunciado pela Intel a fabricação de um processador quântico, em fase de teste, de 49 qubits supercondutores. A tecnologia utilizada atualmente para realizar os qubits são as junções de Josephson, consistindo em uma camada fina de óxido entre dois fios de alumínio. A Intel também pesquisa qubits de spin em silício.

Fonte:

<https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/>

<https://newsroom.intel.com/press-kits/quantum-computing/>

Microsoft

A Microsoft vem empregando esforços na elaboração de qubits topológicos por meio de férmions de Majorana. Além disso, foi lançado em 11 de dezembro de 2017 o Microsoft Quantum Development Kit, contendo a linguagem de programação Q# dedicada para Computação Quântica, e acompanhada de simuladores.

Fonte:

<https://www.microsoft.com/en-us/quantum/technology>

<https://cloudblogs.microsoft.com/quantum/2017/12/11/announcing-microsoft-quantum-development-kit/>

Rigetti

A Startup Rigetti disponibiliza um processador quântico de 19 qubits e um ambiente de desenvolvimento, chamado Forest, para programação quântica. A empresa também disponibiliza um simulador, chamado Quantum Virtual Machine, de 26 qubits. A Rigetti também desenvolveu

a linguagem open source Quil, que baseia-se em um modelo computacional clássico/quântico com memória compartilhada, e trabalha em uma biblioteca python para programação quântica, a pyQuil.

Fonte:

<https://www.rigetti.com/>

<https://www.rigetti.com/forest>

<https://github.com/rigetticomputing/pyQuil>

Outras Startups e Empresas



Figura 4.4: Startups e empresas pioneiras em Computação Quântica.
Fonte: Gartner [4].

Capítulo 5

Computação Quântica com IBM Quantum Experience

5.1 IBM Quantum Experience

A IBM Research disponibiliza um computador de 5 qubits e um de 16 qubits acessíveis pela nuvem (IBM cloud). O acesso aos computadores se dá por meio do QISKit – Quantum Information Software Kit – um pacote de software para python. É possível também acessar o computador de 5 qubits por uma interface gráfica no navegador (Composer). Os detalhes serão vistos na seção 5.1.2.

5.1.1 Computadores Disponíveis

Os computadores quânticos disponíveis para acesso na nuvem são listados abaixo.

Nome	# qubits	Status
IBM Q 5 Yorktown (ibmqx2)	5 qubits	manutenção
IBM Q 5 Tenerife (ibmqx4)	5 qubits	disponível (Composer/QISKit)
IBM Q 16 Rueschlikon (ibmqx5)	16 qubits	disponível (QISKit)

Tabela 5.1: Processadores quânticos da IBM disponíveis para uso pela nuvem. Informação sobre o status dos computadores visualizada em maio de 2018.

Pode-se consultar informações sobre esses processadores no GitHub. Os links são disponibilizados abaixo.

Informação sobre os processadores:

<https://github.com/QISKit/qiskit-backend-information/tree/master/backends>

IBM Q 5 Yorktown (ibmqx2):

<https://github.com/QISKit/qiskit-backend-information/tree/master/backends/yorktown/V1>

IBM Q 5 Tenerife (ibmqx4):

<https://github.com/QISKit/qiskit-backend-information/tree/master/backends/tenerife/V1>

IBM Q 16 Rueschlikon (ibmqx5):

<https://github.com/QISKit/qiskit-backend-information/tree/master/backends/rueschlikon/V1>

5.1.2 Como Programar

A programação é feita por uma interface gráfica, como ilustrado na figura 5.1. Além disso, é possível realizar a programação em modo texto (figura 5.2) por código OpenQASM – Open Quantum Assembly Language. Há opções de simular e de executar o algoritmo no computador quântico. A página do editor de algoritmos da IBM Quantum Experience é acessível pelo link:

<https://quantumexperience.ng.bluemix.net/qx/editor>

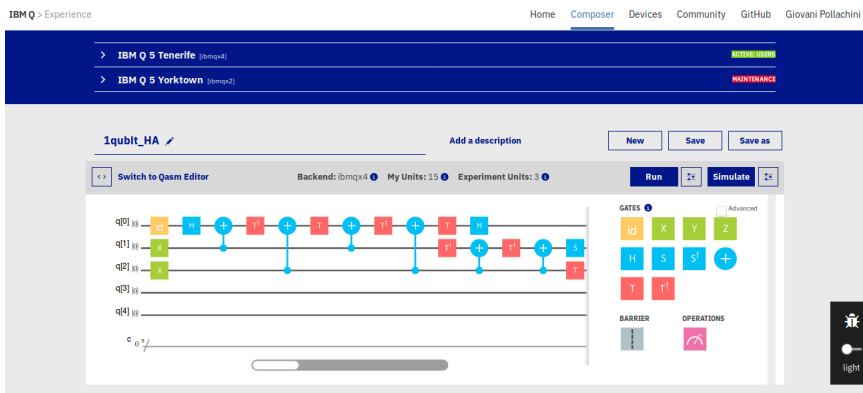


Figura 5.1: Interface gráfica para programação quântica no IBM Quantum Experience.

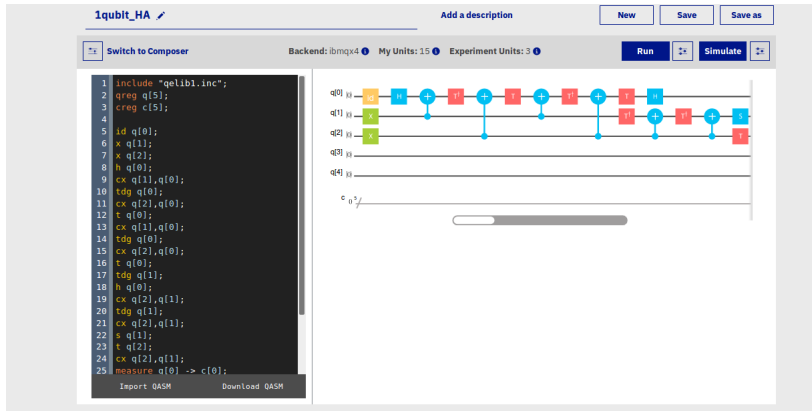


Figura 5.2: Interface para programação em OpenQASM (modo texto) na IBM Quantum Experience.

No IBM Quantum Experience, os qubits são dispostos de acordo com a convenção apresentada na figura .

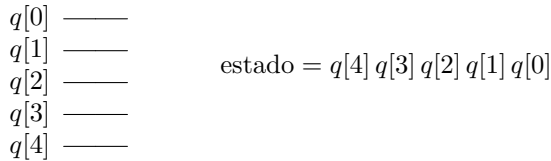


Figura 5.3: Notação utilizada pelo IBM Quantum Experience.

As portas lógicas quânticas disponíveis para uso são apresentadas na figura 5.4. É possível utilizar as portas X, Y, Z de Pauli, a porta de Hadamard H , as portas de fase $S, S^\dagger, T, T^\dagger$. A única porta de dois qubits disponível é a CNOT.



Figura 5.4: Portas lógicas quânticas disponíveis para uso no IBM Quantum Experience.

A porta CNOT não admite controle e alvo em qualquer par de qubits. As combinações possíveis são aquelas em que os dois qubits estão conectados por barramentos supercondutores. As opções são expostas na figura 5.5.

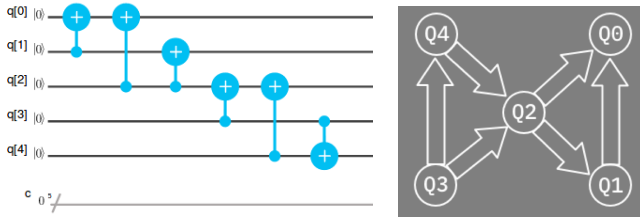


Figura 5.5: Portas CNOT que podem ser implementadas diretamente no IBM Q 5 Tenerife (ibmqx4).

É possível realizar outras portas CNOT em função das CNOTs nativas. Para tanto, faz-se uso das identidades de circuito contidas na seção 3.5. Em especial, são úteis as proposições 3.11 e 3.12.

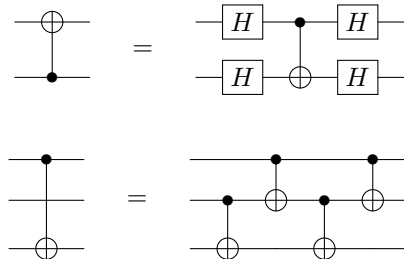


Figura 5.6: Maneiras de se obter CNOTs não nativas no computador IBM Q 5 Tenerife (ibmqx4).

A porta SWAP também pode ser realizada em função de CNOTs nativas. Usam-se as proposições 3.11 e 3.10 da seção 3.5 sobre identidade de circuitos.

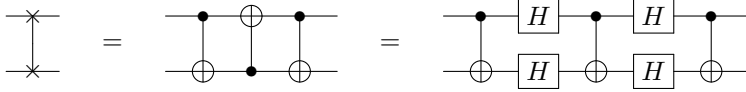


Figura 5.7: Maneira de se obter SWAP em função de CNOTs nativas no computador IBM Q 5 Tenerife (ibmqx4).

A porta Toffoli não é nativa no computador IBM Q 5 Tenerife (ibmqx4). Para realizá-la, pode-se aplicar a proposição 3.16. Essa proposição fornece a seguinte construção.

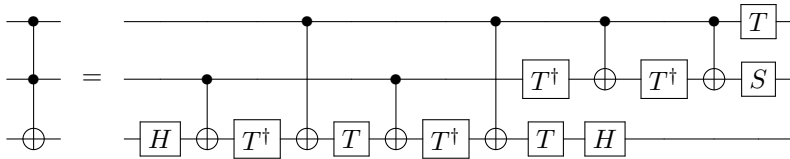


Figura 5.8: Maneira de se obter a porta Toffoli em função de operações nativas no computador IBM Q 5 Tenerife (ibmqx4).

A programação nos computadores quânticos também pode ser feita em python por meio dos pacotes QISKit. O acesso ao computador IBM Q 16 Rueschlikon, de 16 qubits, é feito apenas dessa forma. Um tutorial de QISKit pode ser encontrado no link abaixo.

<https://developer.ibm.com/open/videos/qiskit-quantum-computing-tech-talk/>

5.1.3 Informativos e Guias de Usuário

User Guides

A página do IBM Quantum Experience fornece guias para usuários. Os links estão dispostos a seguir.

Beginner's Guide:

<https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=beginners-guide&page=introduction>

Full User Guide:

<https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=full-user-guide&page=introduction>

User Guides no GitHub:

<https://github.com/QISKit/ibmqx-user-guides>

Informativos

A página da IBM também fornece material informativo contendo palestras e vídeos curtos sobre Computação Quântica.

<http://www.research.ibm.com/ibm-q/learn/what-is-quantum-computing/>

5.2 Circuito Quantum Half Adder

Nesta seção é apresentado um projeto de somador quântico análogo ao somador clássico *half adder*, apresentado no apêndice A.12. Após o projeto, procedeu-se à simulação no IBM Quantum Experience e à execução no computador IBM Q 5 Tenerife (ibmqx4).

5.2.1 Projeto do Circuito

O circuito deve realizar um somador half adder para entradas na base computacional. A tabela verdade do half adder é apresentada a seguir.

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Tabela 5.2: Tabela verdade para o meio somador clássico (half adder). As entradas são simbolizadas por a e b e as saídas são s (sum) e c (carry out).

As equações booleanas para o bit de soma e o bit “vai um” (carry) são

$$\begin{aligned}s &= a \oplus b \\ c &= a \cdot b,\end{aligned}$$

em que \oplus é a operação XOR (ou exclusivo) e \cdot é a operação AND.

Já que há 2 entradas e 2 saídas, é possível tentar realizar esse circuito com apenas 2 qubits. No entanto, há duas entradas distintas $ab = 10$ e $ba = 01$ que fornecem o mesmo resultado $sc = 10$, de forma que a função booleana não é reversível. É necessário, então, usar pelo menos um qubit

de trabalho. Um circuito quântico capaz de realizar as operações desejadas é dado na figura 5.9.

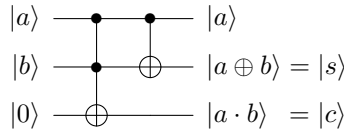


Figura 5.9: Esquemático do quantum half adder, circuito quântico que se comporta, na base computacional, como o Half Adder clássico.

O circuito da figura 5.9 foi adaptado para implementação na plataforma IBM Quantum Experience. A porta Toffoli não é nativa no sistema, e deve ser realizada com as outras portas lógicas quânticas; uma maneira de se fazer isso é por meio da identidade de circuitos apresentada na proposição 3.16.

Há uma limitação com relação ao uso das portas CNOT. As maneiras possíveis de se incluir uma porta CNOT no circuito são mostradas na figura 5.5. Portanto, deve-se utilizar técnicas para construir outras portas CNOT em função das portas nativas. Algumas técnicas são apresentadas na figura 5.6. Alternativamente, pode-se tentar mapear os qubits de forma a conseguir realizar o circuito utilizando as CNOTs nativas.

Outro detalhe é que o sistema inicializa os qubits no estado $|0\rangle$, portanto para aplicar outras entradas é necessário modificar esses estados $|0\rangle$ com portas lógicas. Para inicializar com o estado $|1\rangle$, por exemplo, deve-se aplicar uma porta X ao $|0\rangle$.

Assim, o circuito implementado na plataforma IBM Quantum Experience ficou como disposto na figura 5.10.

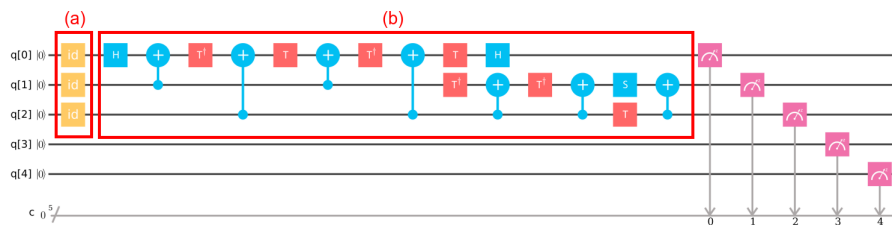


Figura 5.10: Circuito Quantum Half-Adder implementado na plataforma IBM Quantum Experience. (a) Preparação do estado inicial (substituir Id por X para preparar estados $|1\rangle$). (b) Circuito adaptado. Mapeamento de qubits: $q[0] = |c\rangle$ (ancilla), $q[1] = |b\rangle$ e $q[2] = |a\rangle$.

5.2.2 Simulação do Circuito no IBM QX

Foram realizadas simulações do circuito 5.10 para todas as 4 combinações de entradas na base computacional. Para cada entrada, a simulação consiste na execução do circuito por 100 vezes, obtendo-se um gráfico de barras com a contagem dos resultados na base computacional. O número de disparos padrão é 100, mas outras configurações são possíveis.

O resultado das simulações encontra-se na figura 5.11.

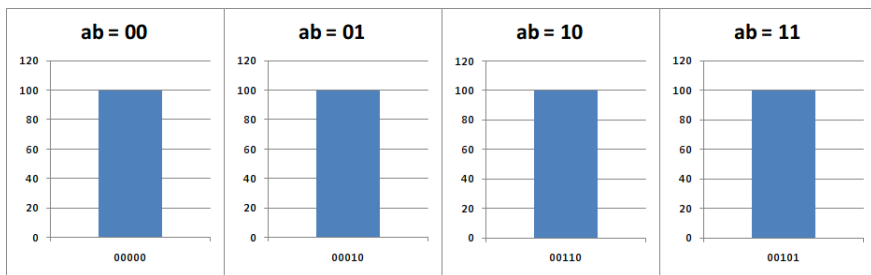


Figura 5.11: Simulação do circuito Quantum Half Adder na plataforma IBM Quantum Experience. Número de disparos para cada entrada: 100.

Portanto, os resultados da simulação podem ser reescritos na tabela abaixo. Pode-se perceber que esses resultados correspondem ao disposto na tabela verdade 5.2.

Entrada $ xxab0\rangle$	Saída $ xxasc\rangle$
000	000
010	010
100	110
110	101

Tabela 5.3: Resultados da simulação do circuito quantum half adder.

5.2.3 Execução do Circuito no IBM QX

O circuito foi importado para o computador IBM Q 5 Tenerife (ibmqx4) por meio da interface IBM Quantum Experience. Foi realizada uma execução para cada entrada diferente, e cada execução consiste em 1024 disparos do circuito (esse número é configurável, podendo-se escolher entre as opções: 1, 1024, 4098 e 8192). Os resultados encontram-se na figura 5.12.

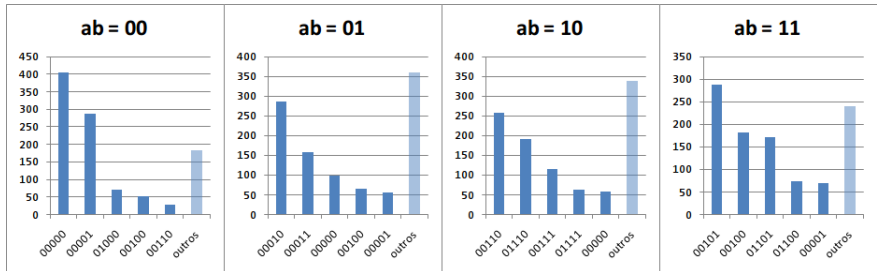


Figura 5.12: Execução do circuito quantum half adder no computador IBM Q 5 Tenerife (ibmqx4). Número de disparos em cada execução: 1024.

Percebe-se que os estados com maior número de contagens correspondem aos estados esperados como resposta para o circuito. Esses estados estão resumidos na tabela 5.4.

Entrada $ xxab0\rangle$	Saída $ xxasc\rangle$	Concordância com simulação ^(*)
00000	00000	$\frac{404}{1024} = 39,5\%$
00010	00010	$\frac{285}{1024} = 27,8\%$
00100	00110	$\frac{258}{1024} = 25,2\%$
00110	00101	$\frac{287}{1024} = 28,0\%$

Tabela 5.4: Execução do circuito quantum half adder no computador IBM Q 5 Tenerife (ibmqx4). (*) Percentual de disparos em que o circuito se comporta como projetado.

Apesar de o estado esperado aparecer com maior frequência, a quantidade de resultados espúrios impede que o algoritmo se comporte como projetado de maneira satisfatória. Esses resultados indicam que é necessário, para a tecnologia atual, utilizar correção de erros no projeto de algoritmos quânticos.

Capítulo 6

Protocolos e Algoritmos Quânticos

Neste capítulo serão abordados alguns protocolos e algoritmos quânticos conhecidos na literatura. Quando possível, é feita uma comparação com os algoritmos clássicos conhecidos.

6.1 Codificação Superdensa

A codificação superdensa é um protocolo que envolve duas partes, Alice e Bob¹, que queiram se comunicar trocando bits. Alice quer enviar bits de mensagem para Bob.

Não é possível, classicamente, codificar 2 bits de mensagem em 1 bit transmitido, já que só há $2^1 = 2$ palavras código – as palavras 0 e 1 – e $2^2 = 4$ palavras de mensagem que podem ser enviadas – as palavras 00, 01, 10 e 11. No entanto, é possível codificar 2 bits de mensagem em 1 qubit transmitido, e é essa a função do *circuito de codificação superdensa*.

Uma referência para este conteúdo é [15], seção 2.3, p.97-98.

6.1.1 Visão geral

A codificação superdensa envolve o compartilhamento prévio de um par de qubits emaranhados, no estado de Bell $|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$. Esse

¹Em Teoria da Informação e Criptografia Quântica, é uma convenção tácita rotular os dois lados da comunicação por Alice e Bob!

estado independe da mensagem que Alice quer enviar a Bob, e pode ter sido distribuído por uma fonte externa de pares emaranhados.

Assim, o primeiro qubit está com Alice e o segundo, com Bob. Alice pode realizar operações em seu qubit em função dos bits de mensagem que ela quer enviar. Após as operações, ela envia seu qubit a Bob, que passa a estar em posse dos dois qubits. Bob pode medi-los de maneira a obter os bits de mensagem.

6.1.2 Circuito

O circuito completo para a codificação superdensa é representado abaixo. Seu funcionamento detalhado será abordado a seguir.

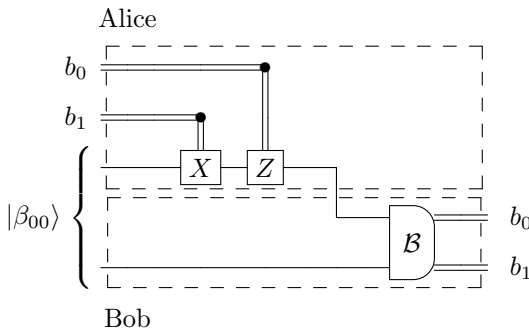


Figura 6.1: Circuito completo para Codificação Superdensa. A medição na base de Bell pode ser realizada pelo circuito da figura 6.3.

6.1.3 Funcionamento Detalhado

Setup

Num primeiro momento, Alice e Bob compartilham o estado de Bell

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Usamos os rótulos A para o primeiro qubit (da Alice) e B para o segundo qubit (que está com Bob).

Codificação - Alice

Se forem realizadas as operações que constam na tabela 6.1, Alice conseguirá 4 estados da base de Bell distintos em função dos bits de mensagem.

Mensagem	Operação	Resultado
00	I_A	$\frac{ 00\rangle+ 11\rangle}{\sqrt{2}} = \beta_{00}\rangle$
01	Z_A	$\frac{ 00\rangle- 11\rangle}{\sqrt{2}} = \beta_{01}\rangle$
10	X_A	$\frac{ 01\rangle+ 10\rangle}{\sqrt{2}} = \beta_{10}\rangle$
11	$iY_A = Z_A X_A$	$\frac{ 01\rangle- 10\rangle}{\sqrt{2}} = \beta_{11}\rangle$

Tabela 6.1: Tabela de codificação da mensagem para a Codificação Superdensa.

Observa-se que $ZX = iY$ pela identidade de circuitos dada na proposição 3.3. Os estados de Bell $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, $|\beta_{10}\rangle$ e $|\beta_{11}\rangle$ formam uma base para o espaço de 2 qubits (seção 2.3). Observa-se também que, se a mensagem é b_0b_1 , então quando $b_1 = 1$ aplica-se X_A e quando $b_0 = 1$, aplica-se Z_A . Dessa forma, para todos os valores da mensagem b_0b_1 , pode-se escrever a operação no qubit A por $Z_A^{b_0} X_A^{b_1}$. Portanto, a operação que Alice deve fazer em seu qubit pode ser representada pelo seguinte circuito controlado por cbits:

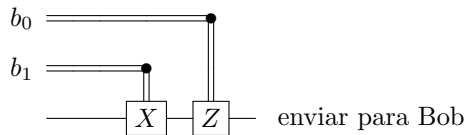


Figura 6.2: Operações que Alice deve fazer em seu qubit antes de enviá-lo para Bob. O circuito realiza a operação $Z_A^{b_0} X_A^{b_1}$.

Decodificação - Bob

Alice envia, então, seu qubit a Bob, que realiza uma medida na base de Bell. Bob consegue distinguir em qual estado o par de qubits se encontra com essa medida, e, consequentemente, consegue saber quais bits de mensagem foram enviados: se o resultado for $|\beta_{b_0b_1}\rangle$, então a mensagem é b_0b_1 .

A medição na base de Bell pode ser realizada em função da medição na base computacional pelo seguinte circuito:

6.2.1 Visão Geral

Como na codificação superdensa, o circuito de teletransporte necessita do compartilhamento prévio de dois qubits emaranhados no estado de Bell $|\beta_{00}\rangle$, que independe do qubit que Alice quer enviar a Bob.

Além desse par compartilhado, Alice tem um qubit em um estado $|\psi\rangle$ desconhecido. Alice realiza determinadas operações nos seus dois qubits e realiza medidas na base computacional. Ao medir seus dois qubits, ela obtém informação clássica (cbits), e envia-as a Bob.

Bob recebe os cbits e, em função do resultado, realiza algumas operações em seu qubit (o segundo qubit do par emaranhado compartilhado previamente). Essas operações o ajudam a recuperar o estado do qubit que Alice queria enviar, completando a tarefa.

Nesse processo, o qubit $|\psi\rangle$ de Alice tem seu estado destruído pela medida, mas reaparece no qubit de Bob por causa do emaranhamento, restando apenas realizar uma correção em função do resultado da medida de Alice.

6.2.2 Circuito

O circuito abaixo realiza o teletransporte do estado de 1 qubit de Alice para Bob utilizando apenas o envio de 2 cbits. O funcionamento detalhado do Circuito de Teletransporte será visto adiante.

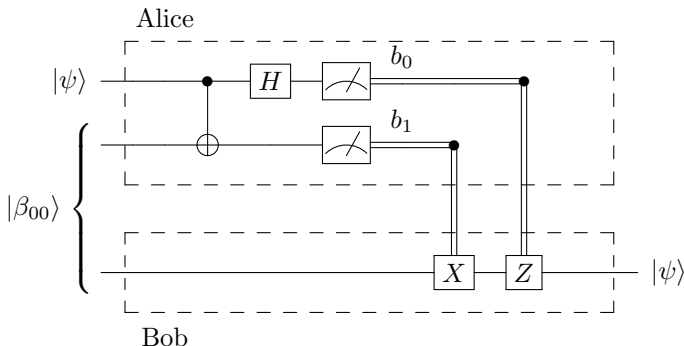


Figura 6.4: Circuito de Teletransporte completo.

6.2.3 Funcionamento Detalhado

Setup

Alice e Bob previamente compartilham o estado de Bell

$$|\beta_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}.$$

Alice também possui um qubit $|\psi\rangle = a|0\rangle + b|1\rangle$ em um estado não necessariamente conhecido por ela. Sua intenção é que Bob tenha uma “cópia” do estado desse qubit. Usando-se os rótulos A_1 e A_2 para os qubits de Alice e B para o de Bob, o estado do sistema completo pode ser escrito como

$$\begin{aligned} |\psi_0\rangle &= |\psi\rangle_{A_1} |\beta_{00}\rangle_{A_2 B} \\ &= (a|0\rangle_{A_1} + b|1\rangle_{A_1}) \frac{1}{\sqrt{2}} (|0\rangle_{A_2} |0\rangle_B + |1\rangle_{A_2} |1\rangle_B) \\ &= \frac{a}{\sqrt{2}} |0\rangle_{A_1} |0\rangle_{A_2} |0\rangle_B + \frac{a}{\sqrt{2}} |0\rangle_{A_1} |1\rangle_{A_2} |1\rangle_B \\ &\quad + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |0\rangle_{A_2} |0\rangle_B + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |1\rangle_{A_2} |1\rangle_B \\ &= \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |0\rangle_{A_2} \right) |0\rangle_B \\ &\quad + \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |1\rangle_{A_2} \right) |1\rangle_B. \end{aligned}$$

Preparação - Alice

Alice realiza as operações CNOT nos dois qubits e Hadamard em A_1 , e o sistema completo passa a ficar no estado:

$$\begin{aligned} |\psi_1\rangle &= \text{CNOT}_{A_1, A_2} |\psi_0\rangle \\ &= \text{CNOT}_{A_1, A_2} \left[\left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |0\rangle_{A_2} \right) |0\rangle_B \right. \\ &\quad \left. + \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |1\rangle_{A_2} \right) |1\rangle_B \right] \\ &= \left(\frac{a}{\sqrt{2}} \text{CNOT}_{A_1, A_2} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} \text{CNOT}_{A_1, A_2} |1\rangle_{A_1} |0\rangle_{A_2} \right) |0\rangle_B \\ &\quad + \left(\frac{a}{\sqrt{2}} \text{CNOT}_{A_1, A_2} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} \text{CNOT}_{A_1, A_2} |1\rangle_{A_1} |1\rangle_{A_2} \right) |1\rangle_B \\ &= \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |1\rangle_{A_2} \right) |0\rangle_B \\ &\quad + \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |0\rangle_{A_2} \right) |1\rangle_B \end{aligned}$$

$$\begin{aligned}
|\psi_2\rangle &= H_{A_1} |\psi_1\rangle \\
&= H_{A_1} \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |1\rangle_{A_2} \right) |0\rangle_B \\
&\quad + H_{A_1} \left(\frac{a}{\sqrt{2}} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} |1\rangle_{A_1} |0\rangle_{A_2} \right) |1\rangle_B \\
&= \left(\frac{a}{\sqrt{2}} H_{A_1} |0\rangle_{A_1} |0\rangle_{A_2} + \frac{b}{\sqrt{2}} H_{A_1} |1\rangle_{A_1} |1\rangle_{A_2} \right) |0\rangle_B \\
&\quad + \left(\frac{a}{\sqrt{2}} H_{A_1} |0\rangle_{A_1} |1\rangle_{A_2} + \frac{b}{\sqrt{2}} H_{A_1} |1\rangle_{A_1} |0\rangle_{A_2} \right) |1\rangle_B \\
&= \left(\frac{a}{2} (|0\rangle_{A_1} + |1\rangle_{A_1}) |0\rangle_{A_2} + \frac{b}{2} (|0\rangle_{A_1} - |1\rangle_{A_1}) |1\rangle_{A_2} \right) |0\rangle_B \\
&\quad + \left(\frac{a}{2} (|0\rangle_{A_1} + |1\rangle_{A_1}) |1\rangle_{A_2} + \frac{b}{2} (|0\rangle_{A_1} - |1\rangle_{A_1}) |0\rangle_{A_2} \right) |1\rangle_B \\
&= |00\rangle_{A_1 A_2} \left(\frac{a}{2} |0\rangle_B + \frac{b}{2} |1\rangle_B \right) + |01\rangle_{A_1 A_2} \left(\frac{b}{2} |0\rangle_B + \frac{a}{2} |1\rangle_B \right) \\
&\quad + |10\rangle_{A_1 A_2} \left(\frac{a}{2} |0\rangle_B - \frac{b}{2} |1\rangle_B \right) + |11\rangle_{A_1 A_2} \left(-\frac{b}{2} |0\rangle_B + \frac{a}{2} |1\rangle_B \right)
\end{aligned}$$

Alice realiza, então, a medida dos seus dois qubits na base computacional. Essa medida faz com que o sistema total encontre-se no estado

$$|\psi_3\rangle = |b_0 b_1\rangle_{A_1 A_2} |\psi_3\rangle_B \quad .$$

em que o resultado da medida é $b_0 b_1$. O qubit que está com Bob passa a ficar no estado $|\psi_3\rangle_B$, que depende do valor da medida. As opções possíveis são listadas na tabela a seguir.

Resultado da medida	Estado do qubit B
00	$a 0\rangle + b 1\rangle$
01	$b 0\rangle + a 1\rangle$
10	$a 0\rangle - b 1\rangle$
11	$-b 0\rangle + a 1\rangle$

Tabela 6.2: Possíveis resultados da medição de Alice para o Circuito de Teletransporte.

O cálculo da primeira linha da tabela 6.2 é exemplificado a seguir. Caso o resultado da medida tenha sido 00, o estado do sistema total $|\psi_3\rangle$ é obtido pela projeção $|00\rangle\langle 00|_{A_1 A_2}$ seguida de uma normalização do vetor resultante. É conveniente lembrar que $|a|^2 + |b|^2 = 1$ pela normalização do estado $|\psi\rangle = a |0\rangle + b |1\rangle$ do início do algoritmo.

$$|00\rangle\langle 00|_{A_1 A_2} |\psi_2\rangle = |00\rangle_{A_1 A_2} \left(\frac{a}{2} |0\rangle_B + \frac{b}{2} |1\rangle_B \right)$$

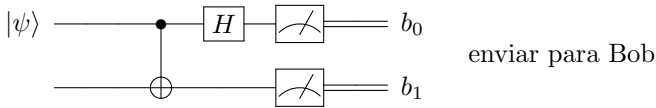


Figura 6.5: Alice realiza operações em seus qubits e envia informação clássica para Bob. O estado do qubit que ela deseja enviar se perde no processo de medição.

$$\begin{aligned}
 |\psi_2\rangle &= \frac{|00\rangle_{A_1 A_2} \left(\frac{a}{2} |0\rangle_B + \frac{b}{2} |1\rangle_B \right)}{\sqrt{\left| \frac{a}{2} \right|^2 + \left| \frac{b}{2} \right|^2}} \\
 &= \frac{|00\rangle_{A_1 A_2} \left(\frac{a}{2} |0\rangle_B + \frac{b}{2} |1\rangle_B \right)}{\frac{1}{2} \sqrt{|a|^2 + |b|^2}} \\
 &= |00\rangle_{A_1 A_2} (a |0\rangle_B + b |1\rangle_B) .
 \end{aligned}$$

Os outros estados da tabela são obtidos com contas similares.

Agora, Alice pode enviar o resultado da medida para Bob e o qubit em posse dele ficará no estado correspondente na tabela 6.2.

Processamento final - Bob

Nesse ponto, Bob já tem conhecimento do resultado da medida de Alice e o estado do seu qubit corresponde à entrada correspondente na tabela 6.2. Para recuperar o estado $a|0\rangle + b|1\rangle$, Bob deve fazer algumas operações para corrigir o estado do seu qubit, em função do valor da medida informado a ele.

Se a medida for 00, seu qubit está em $a|0\rangle + b|1\rangle$ e nada precisa ser feito. Se a medida resultou em 01, seu qubit está em $b|0\rangle + a|1\rangle$; nesse caso, é possível perceber que a porta X fornece novamente o estado desejado $b|1\rangle + a|0\rangle$. Considerando-se todos os resultados possíveis da medida, monta-se a correção necessária para cada caso, conforme disposto na tabela 6.3.

Medida	Estado do qubit B	Aplicar operações	Estado final
00	$a 0\rangle + b 1\rangle$	I_B	$a 0\rangle + b 1\rangle = \psi\rangle$
01	$b 0\rangle + a 1\rangle$	X_B	$a 0\rangle + b 1\rangle = \psi\rangle$
10	$a 0\rangle - b 1\rangle$	Z_B	$a 0\rangle + b 1\rangle = \psi\rangle$
11	$-b 0\rangle + a 1\rangle$	$Z_B X_B$	$a 0\rangle + b 1\rangle = \psi\rangle$

Tabela 6.3: Correções aplicadas por Bob para obter o estado original do qubit de Alice no Circuito de Teletransporte.

Essas operações, como na codificação superdensa, podem ser resumidas na operação controlada classicamente $Z_B^{b_0} X_B^{b_1}$, em que $b_0 b_1$ é o resultado da medida. Dessa forma, pode-se representar o processamento de Bob pelo circuito a seguir.

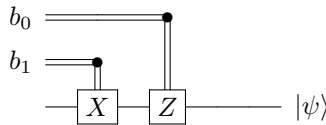


Figura 6.6: Bob recebe os bits enviados por Alice e, em função dos valores recebidos, realiza um processamento final em seu qubit, recuperando o estado $|\psi\rangle$ que Alice tinha e que pretendia enviar.

Com isso, independente do resultado da medida, Bob tem, ao final, o estado $|\psi\rangle$ que Alice queria transmitir.

6.3 Oráculos Quânticos

Os oráculos são funções booleanas $f: \{0,1\}^n \rightarrow \{0,1\}$ consideradas como “caixas pretas”. Dado um vetor de bits x , o oráculo clássico fornece $f(x)$. Alguns problemas computacionais são escritos em termos de oráculos, como o problema de Deutsch-Jozsa, o problema de Simon e o problema de busca de Grover. Para abordar esses problemas, é necessário definir uma versão quântica desse oráculo, o que será abordado nesta seção.

Há duas maneiras de se escrever um análogo quântico ao oráculo clássico: o *oráculo XOR* e o *oráculo de fase*.

6.3.1 Oráculo XOR

O *oráculo XOR* é uma operação unitária que realiza a função booleana f por meio de um bit extra, que sinaliza as entradas em que f vale 1.

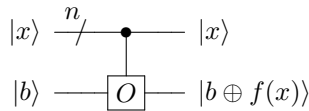


Figura 6.7: Oráculo quântico XOR. O comportamento na base computacional está descrito no circuito. O rótulo n no primeiro fio representa n qubits. O controle inverte o $n + 1$ -ésimo qubit quando os n primeiros qubits $|x\rangle$ da entrada cumprem $f(x) = 1$.

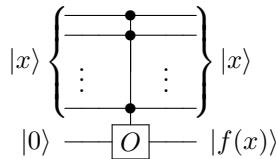


Figura 6.8: Oráculo quântico XOR. Comportamento quando $|b\rangle = |0\rangle$ e quando $|x\rangle$ é vetor da base computacional.

O oráculo XOR pode ser generalizado para funções $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ ou, ainda, para funções $f: \{0, 1\}^n \rightarrow \{0, 1\}^m$.

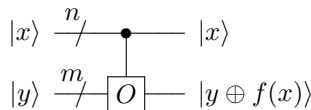


Figura 6.9: Oráculo quântico XOR para funções booleanas com entradas de n bits e saídas de m bits. A operação \oplus é a XOR realizada bit a bit.

6.3.2 Oráculo de Fase

O *oráculo de fase* é uma operação unitária que sinaliza as entradas em que f vale 1 introduzindo uma fase de π , isto é, uma multiplicação por -1 .

$$|x\rangle \xrightarrow{n} \boxed{O} \rightarrow (-1)^{f(x)} |x\rangle$$

Figura 6.10: Oráculo quântico de fase. O comportamento na base computacional está descrito no circuito. O rótulo n no primeiro fio representa n qubits. A fase da entrada $|x\rangle$ fica invertida quando $f(x) = 1$.

6.3.3 Construção do Oráculo de Fase usando o Oráculo XOR

Pode-se obter o oráculo de fase a partir do oráculo XOR com o uso do qubit alvo como um qubit auxiliar. Ao usarmos $|-\rangle$ na entrada alvo do oráculo XOR, obtemos, para qualquer estado $|x\rangle$ da base computacional:

$$|x\rangle |-\rangle = |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \xrightarrow{O_{\text{XOR}}} \begin{cases} |x\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |x\rangle |-\rangle & \text{se } f(x) = 0 \\ |x\rangle \frac{|1\rangle - |0\rangle}{\sqrt{2}} = |x\rangle (-|-\rangle) & \text{se } f(x) = 1 \end{cases},$$

o que pode ser resumido por $|x\rangle ((-1)^{f(x)} |-\rangle)$. Além disso, o fator multiplicativo $(-1)^{f(x)}$ pode ser movido para qualquer entrada tensorial por multilinearidade do produto tensorial:

$$|x\rangle \otimes (-1)^{f(x)} |-\rangle = (-1)^{f(x)} |x\rangle \otimes |-\rangle.$$

A figura a seguir ilustra a construção do oráculo de fase.

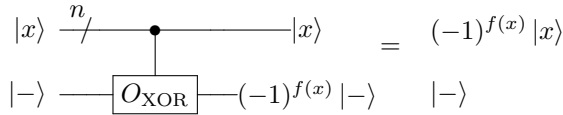


Figura 6.11: Construção do oráculo de fase a partir do oráculo XOR. $|x\rangle$ representa um estado da base computacional.

6.4 Algoritmo de Deutsch-Jozsa

O Algoritmo de Deutsch-Jozsa é um algoritmo quântico projetado para resolver o Problema de Deutsch-Jozsa. Esse problema não tem especial ênfase em aplicações, mas torna-se laboratório interessante para investigar técnicas e possíveis vantagens da Computação Quântica. As principais referências para esta seção são o livro [15], seção 1.4.4, p.34-36, e as videoaulas U2.2, subunidade SU2 de [19].

6.4.1 Problema de Deutsch-Jozsa

Antes de enunciar o problema de Deutsch-Jozsa é conveniente escrever algumas definições.

Definição 6.1 (Função constante e função balanceada).

A função booleana $f: \{0,1\}^n \rightarrow \{0,1\}$ é dita *constante* se f assume o mesmo valor em todas as entradas:

$$\begin{aligned} f(x) &= 0, \quad \forall x \in \{0,1\}^n \quad \text{ou} \\ f(x) &= 1, \quad \forall x \in \{0,1\}^n. \end{aligned}$$

A função f é dita *balanceada* se admite o valor 0 em metade das suas entradas e admite 1 na metade complementar das entradas.

Exemplo 6.2. A função booleana $f(x) = 1$ é constante.

Exemplo 6.3. Denote $x \in \{0,1\}^n$ por $x = x_{n-1} \dots x_1 x_0$. A função booleana $f(x) = x_0$ é balanceada, pois para exatamente metade das entradas x tem-se $x_0 = 0$ e para a outra metade, tem-se $x_0 = 1$.

Exemplo 6.4. Considere a função booleana com entradas de $n = 2$ bits dada por $f(a, b) = a \cdot b$, em que, lembrando, \cdot representa a porta AND. A tabela verdade dessa função é representada abaixo.

a	b	$f(a, b) = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

Essa função não é balanceada nem constante.

O problema desta seção tem o seguinte enunciado.

Problema de Deutsch-Jozsa. *Seja uma função booleana $f: \{0,1\}^n \rightarrow \{0,1\}$ que pode ser apenas ou constante ou balanceada. Decidir se f é constante ou balanceada.*

Deseja-se, dada uma função f considerada como caixa preta, e com o compromisso de ser ou constante ou balanceada, decidir qual dos dois casos mutuamente excludentes é verdadeiro.

6.4.2 Algoritmo de Deutsch-Jozsa

Para resolver o problema de Deutsch-Jozsa com um algoritmo quântico, é necessário ter uma versão quântica da função booleana f , dada como oráculo, isto é, dada como uma caixa preta em que não se pode visualizar a subrotina que calcula f .

Considere que f seja dada por meio do oráculo de fase (como na seção 6.3.2). O algoritmo de Deutsch-Jozsa para decidir se f é constante ou balanceada é dado pelo procedimento abaixo.

Algoritmo de Deutsch-Jozsa

Entrada: $O_F(f) = O$ (oráculo de fase associado à função booleana f)

Procedimento:

- | | | |
|----------|--|--|
| etapa 0: | $ 0\rangle^{\otimes n}$ | preparação do estado inicial |
| etapa 1: | $ +\rangle^{\otimes n}$ | superposição de estados com $H^{\otimes n}$ |
| etapa 2: | $O_F +\rangle^{\otimes n}$ | aplicação de f (oráculo de fase) |
| etapa 3: | $\langle + ^{\otimes n} O_F +\rangle^{\otimes n}$ | testar para $ +\rangle^{\otimes n}$ (base girada \mathcal{X}) |

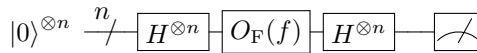
Saída: Probabilidade da medida de $|\psi_2\rangle$ resultar em $|+\rangle^{\otimes n}$ é

$$P = \begin{cases} 1 & \text{se } f \text{ é constante} \\ 0 & \text{se } f \text{ é balanceada} \end{cases}$$

Portanto, se o estado após a medida na base \mathcal{X} for $|+\rangle^{\otimes n}$, então decide-se que f é constante. E se o estado após a medida for qualquer outro, decide-se que f é balanceada.

Circuito

Notação compacta:



Notação expandida:

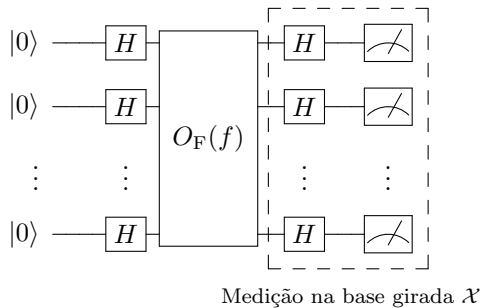


Figura 6.12: Algoritmo de Deutsch-Jozsa.

Observação 6.5. A porção destacada na figura 6.12 corresponde à medição na base \mathcal{X} feita a partir da medição na base computacional. De fato, o

operador de Hadamard realiza mudança de base de \mathcal{X} (base girada) para \mathcal{I} (base computacional), conforme exemplo 1.23, de forma que o resultado medido na base computacional corresponde a uma medição na base \mathcal{X} . A figura 6.13 ilustra a medição na base girada feita em função da medição na base computacional.



Figura 6.13: Medição na base girada \mathcal{X} realizada em função de medição na base computacional.

Análise detalhada do algoritmo

Na etapa 1, aplica-se H para cada qubit de entrada, resultando em:

$$\begin{aligned}
 |\psi_1\rangle &= H^{\otimes n} |0\rangle^{\otimes n} \\
 &= |+\rangle^{\otimes n} \\
 &= \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \dots \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \right) \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} |x\rangle,
 \end{aligned}$$

em que \mathbb{B}_n representa o conjunto de todas as palavras de n bits. Isto é,

$$\begin{aligned}
 \mathbb{B}_n &= \{0 \dots 00, 0 \dots 01, 0 \dots 10, 0 \dots 11, \dots, 1 \dots 11\} \\
 &= \{0, 1, 2, 3, \dots, 2^n - 1\}.
 \end{aligned}$$

Observação 6.6. Por vezes é útil fazer a identificação entre vetores de bits e números inteiros sem sinal, para simplificar a notação. Por exemplo, $0 = 0 \dots 000$, $1 = 0 \dots 001$, $2 = 0 \dots 010$, $3 = 0 \dots 011$ e assim por diante, até $2^n - 1 = 1 \dots 111$. Essa identificação consta na tabela A.2 do apêndice A.

A aplicação do oráculo na etapa 2 fornece:

$$\begin{aligned}
 |\psi_2\rangle &= O_F |\psi_1\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} O |x\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} (-1)^{f(x)} |x\rangle.
 \end{aligned}$$

Se a função for constante, o fator $(-1)^{f(x)}$ se tornará um sinal global $+$ ou $-$, que essencialmente não altera o estado anterior.

A última etapa consiste na medição na base girada \mathcal{X} . Para realizar essa medida, pode-se aplicar H a todos os qubits e medir na base computacional, como ilustrado na figura 6.12. Calculando a probabilidade de se obter $|+\rangle^{\otimes n}$, consegue-se:

$$\begin{aligned} \langle +|^{\otimes n} |\psi_2\rangle &= \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} \langle y| \right) \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} (-1)^{f(x)} |x\rangle \\ &= \frac{1}{2^n} \sum_{x \in \mathbb{B}_n} \sum_{y \in \mathbb{B}_n} (-1)^{f(x)} \langle y|x\rangle \\ &= \frac{1}{2^n} \sum_{x \in \mathbb{B}_n} \sum_{y \in \mathbb{B}_n} (-1)^{f(x)} \delta_{x,y} \\ &= \frac{1}{2^n} \sum_{x \in \mathbb{B}_n} (-1)^{f(x)} . \end{aligned}$$

Caso a função seja constante, a última equação fornece

$$\frac{1}{2^n} \sum_{x \in \mathbb{B}_n} (-1)^{f(x)} = \pm \frac{1}{2^n} 2^n = \pm 1 .$$

E caso a função seja balanceada, metade das parcelas contribui com 1 e a outra metade com -1 , portanto

$$\frac{1}{2^n} \sum_{x \in \mathbb{B}_n} (-1)^{f(x)} = \frac{1}{2^n} 0 = 0 .$$

A probabilidade P de se obter $|+\rangle^{\otimes n}$ é dada pelo módulo ao quadrado do resultado obtido, logo

$$P = \left| \langle +|^{\otimes n} |\psi_2\rangle \right|^2 = \begin{cases} 1 & \text{se } f \text{ é constante} \\ 0 & \text{se } f \text{ é balanceada} . \end{cases}$$

Dessa forma, decide-se por “ f é constante” se a medida resultar no estado $|+\rangle^{\otimes n}$ e por “ f é balanceada”, se resultar em um estado diferente. Esse teste é realizado no algoritmo por uma mudança de base, realizada pela porta Hadamard, e uma medição na base computacional, como ilustrado nas figuras 6.12 e 6.13.

6.4.3 Algoritmo Clássico

Agora considere o problema de Deutsch-Jozsa no contexto clássico. Tem-se f dada como uma caixa preta e se quer decidir se f é constante

ou balanceada. A seguir serão vistas brevemente as abordagens clássicas determinística e aleatória para o problema.

Algoritmo Clássico Determinístico

A Computação Clássica Determinística é um tipo de computação em que se busca algoritmos que não façam uso de recursos probabilísticos para resolver um problema. Os algoritmos determinísticos são tais que, ao serem executados diversas vezes para uma mesma entrada, produz-se sempre a mesma saída. Para que se resolva o problema nesse tipo de computação, é necessário realizar aplicações sucessivas de f para diversas entradas até se ter certeza de qual opção é válida (se f é constante ou balanceada). Por exemplo², calcula-se $f(0)$, $f(1)$, $f(2)$, \dots e se verifica se $f(1) = f(0)$, $f(2) = f(1)$, \dots ou não. Caso ocorra $f(j) \neq f(i)$, então a opção certa é “ f é balanceada”, e caso isso não ocorra, a opção correta é “ f é constante”.

Para se distinguir com certeza as duas opções, deve-se aplicar f a metade das entradas possíveis mais uma, ou seja, a $2^n/2 + 1$ entradas. Isso porque, na pior das hipóteses, a função era balanceada e, obteve-se um mesmo resultado, por azar, para as $2^n/2$ entradas testadas, impedindo que se faça a escolha com certeza.

Dessa forma, o custo computacional desse algoritmo é de $2^n/2 + 1$ aplicações de f .

Algoritmo Clássico Probabilístico

Um algoritmo probabilístico utiliza a probabilidade como recurso computacional. Para esse tipo de computação, é possível que entradas iguais produzam saídas diferentes, e que a máquina passe por estados diferentes durante a computação, em função de fatores probabilísticos presentes no algoritmo. Nesse contexto, se for permitida uma probabilidade de erro ε na decisão e o uso de sorteios aleatórios em certas etapas, é possível reduzir o custo computacional do algoritmo clássico determinístico.

Primeiramente, permite-se que as entradas i sejam tiradas aleatoriamente, cada uma com mesma probabilidade $p(i) = 1/2^n$. Por exemplo, se f for constante 1 ($f(i) = 1 \forall j$), a probabilidade de resultar 1 é $1 = 100\%$ e a de resultar 0 é $0 = 0\%$. Se f for balanceada, a probabilidade de resultar 1 é $0,5 = 50\%$ e o mesmo vale para o resultado 0. Supõe-se, para simplificar a discussão, que o sorteio das entradas é feito sem memória³, isto é, com chance de se sortear duas entradas iguais.

²Nesta parte, usou-se a notação que confunde uma palavra de bits com sua representação por número inteiro sem sinal. Ver observação 6.6.

³Para um número de bits n grande, esse caso é semelhante ao caso com memória, em que não se permite repetir as entradas no sorteio.

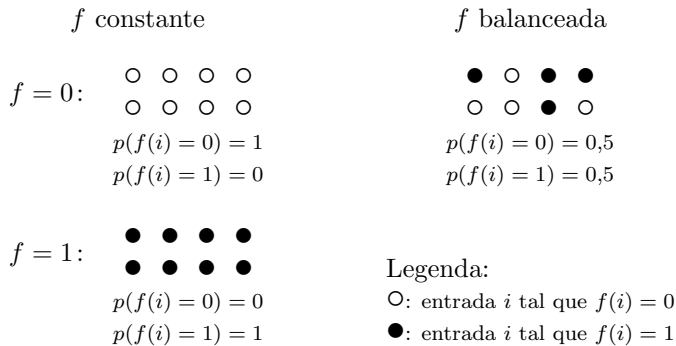


Figura 6.14: Probabilidade de obtenção dos resultados $f(i) = 0$ e $f(i) = 1$ para os casos de f constante e f balanceada ($n = 3$ bits). As entradas são sorteadas aleatoriamente, com igual probabilidade.

A primeira avaliação $f(i_1)$ não traz mais informação para distinguir entre constante e balanceada.

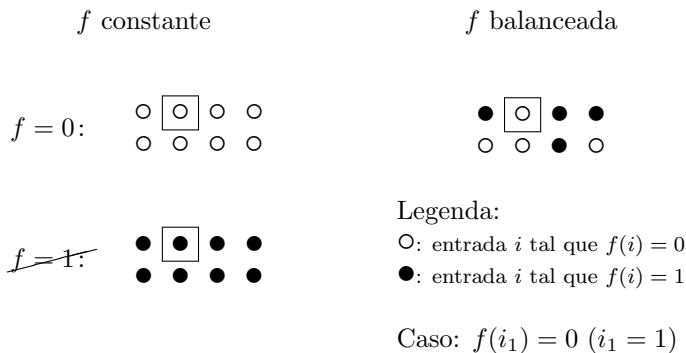


Figura 6.15: Exemplo de sorteio aleatório de uma entrada i_1 e avaliação $f(i_1)$. Se $f(i_1) = 0$, não é possível distinguir ainda se $f = 0$ ou se f é balanceada.

A segunda aplicação, se resultar $f(i_2) \neq f(i_1)$, já resolve com certeza que f é balanceada.

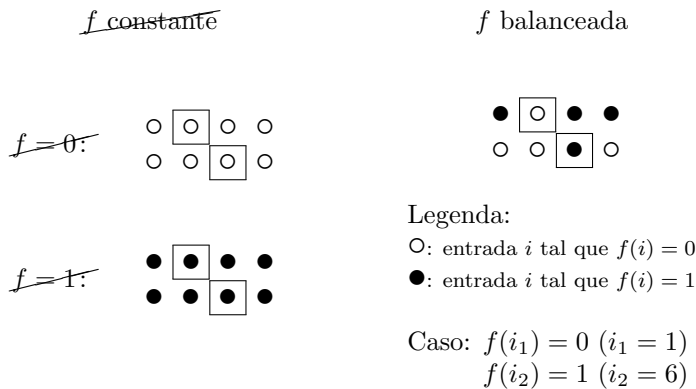


Figura 6.16: Exemplo de sorteio aleatório de duas entradas i_1, i_2 . Se $f(i_1) \neq f(i_2)$, conclui-se que f é balanceada sem probabilidade de erro.

Se o resultado for $f(i_2) = f(i_1)$, tende-se a pensar que f seria constante e a probabilidade de se estar errado é a probabilidade de tirar duas saídas iguais aleatoriamente numa função balanceada, ou seja, $P_e = 1 \cdot 0,5 = 0,5$.

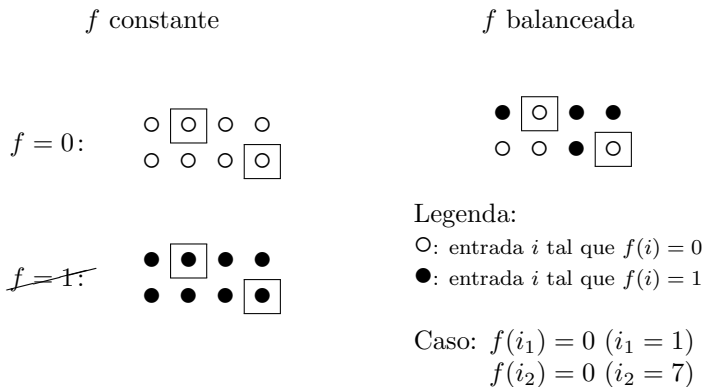


Figura 6.17: Exemplo de sorteio aleatório de duas entradas i_1, i_2 . Caso $f(i_1) = f(i_2)$ e se decida parar o algoritmo, a resposta escolhida seria “ f é constante”, e a probabilidade de erro seria $P_e = 50\%$. Isto é, a probabilidade de, caso a função seja balanceada, ter-se obtido duas entradas iguais, seria $P_e = 50\%$.

Na terceira etapa, caso $f(i_3) \neq f(i_2)$, resolve-se com certeza que f é balanceada e caso $f(i_3) = f(i_2)$, conclui-se pela opção constante com

probabilidade de erro igual a $P_e = 1 \cdot 0,5 \cdot 0,5 = 0,25$, correspondente à probabilidade de que, numa função balanceada, tenha-se o mesmo resultado para 3 entradas sorteadas aleatoriamente com igual probabilidade.

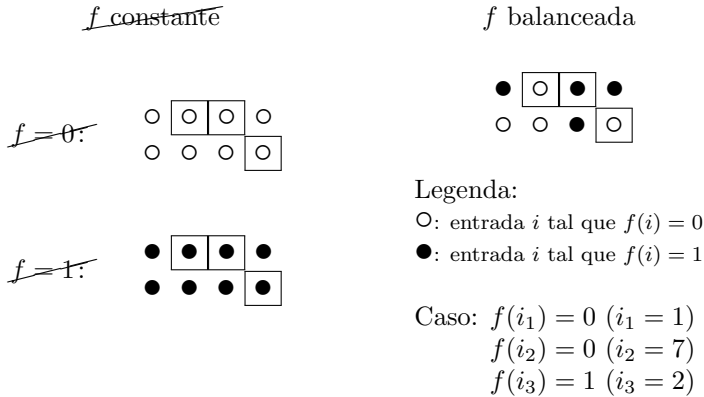


Figura 6.18: Exemplo de sorteio aleatório de três entradas i_1, i_2, i_3 . Caso $f(i_2) \neq f(i_3)$, pode-se concluir que f é balanceada sem chance de erro.

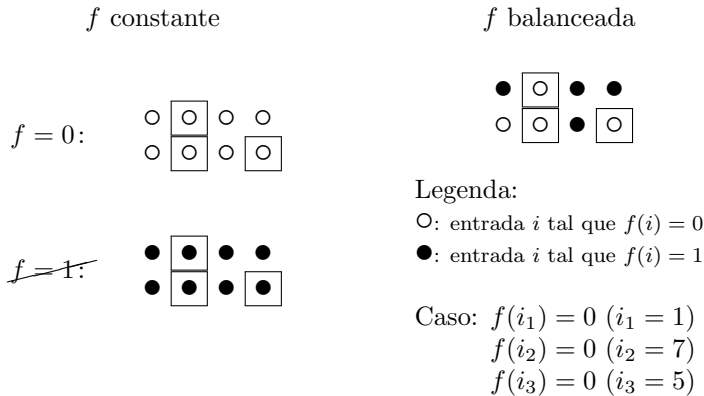


Figura 6.19: Exemplo de sorteio aleatório de três entradas i_1, i_2, i_3 . Caso $f(i_2) = f(i_3)$ e se encerre o algoritmo, escolhe-se a opção “ f é constante” com probabilidade de erro $P_e = 0,25 = 25\%$.

Seguindo essa ideia, na m -ésima aplicação de f , se ocorrer $f(i_m) \neq f(i_{m-1})$, conclui-se com certeza a opção “ f é balanceada” e se $f(i_m) =$

$f(i_{m-1})$, pode-se concluir que “ f é constante” com probabilidade de erro

$$P_e = 1 \cdot 0,5 \cdot \dots \cdot 0,5 = (0,5)^{m-1} = 1/2^{m-1}.$$

Para uma probabilidade de erro $P_e < 1/2$ na decisão, deve-se repetir o algoritmo até que a probabilidade de erro $P_{e,m} = 1/2^{m-1}$ satisfaça

$$\frac{1}{2^{m-1}} < \frac{1}{2} \implies 2^m > 2^2 \implies m > 2 \implies m \geq 3.$$

Se forem $m = 3$ aplicações, a probabilidade de erro será limitada por $\varepsilon = 1/2^{m-1} = 0,25 < 0,5$, como visto anteriormente.

6.4.4 Comparação de Desempenho

A tabela abaixo traz a comparação entre o desempenho dos algoritmos clássico determinístico, clássico probabilístico e quântico.

Algoritmo	Desempenho (# aplicações de f)
Class. Det.	$2^n/2 + 1$
Class. Prob.	3
Quântico	1

Tabela 6.4: Comparação de desempenho entre os algoritmos quântico, clássico determinístico e clássico probabilístico (com probabilidade de erro < 50%) para o problema de Deutsch-Jozsa.

Essa comparação entre o desempenho clássico e quântico, no entanto, não pode ser considerada muito seriamente. Há que se levar em conta que são arquiteturas diferentes: aplicar uma operação f clássica (correspondente a chamar uma subrotina “caixa preta”) e aplicar o oráculo de fase $O_F(f)$ em um circuito quântico são coisas distintas. Não é claro que essas operações têm custo computacional equivalente para que sejam comparadas diretamente como na tabela apresentada. Por outro lado, como comparação simplificada, essa análise serve para se ter uma noção dos ganhos que a Computação Quântica poderia trazer em relação a Computação Clássica.

Em relação ao algoritmo clássico determinístico, o algoritmo quântico apresenta ganho exponencial em desempenho. Já em relação ao algoritmo clássico probabilístico, o desempenho é semelhante.

6.5 Algoritmo de Simon

Assim como no caso do Algoritmo de Deutsch-Jozsa, o Algoritmo de Simon é um algoritmo quântico projetado para resolver o Problema de

Simon. Esse problema também tem propósito de funcionar como laboratório de testes para a Computação Quântica, não apresentando aplicações conhecidas. A principal referência para esta seção são as videoaulas U2.3 da subunidade SU2 disponíveis em [19].

6.5.1 Problema de Simon

O problema de Simon é um problema de promessa, em que é dada uma função booleana $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ que pode ser ou 1-para-1 ou 2-para-1. Esses termos serão definidos e acompanhados de exemplos para melhor entendimento. Em seguida, o enunciado do problema de Simon será escrito formalmente.

Definição 6.7 (Função 1-para-1 e 2-para-1). Seja $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ uma função booleana de n para n bits.

A função f é dita *1-para-1* se é uma bijeção⁴. Nesse caso, isso significa que cada resultado $y \in \{0, 1\}^n$ é obtido por exatamente uma entrada x_1 , ou seja, $f(x_1) = y$. Cada duas entradas distintas $x_1 \neq x_2$ geram resultados diferentes $f(x_1) \neq f(x_2)$.

A função f é dita *2-para-1* se cada resultado $y \in \{0, 1\}^n$ é obtido por exatamente duas entradas x_1 e x_2 , isto é, $f(x_1) = f(x_2) = y$.

O problema de Simon requer um compromisso para a função booleana de entrada. A propriedade que a função deve satisfazer é chamada, no presente trabalho, de *propriedade de Simon*.

Definição 6.8 (Propriedade de Simon). Sejam $f: \{0, 1\}^n \rightarrow \{0, 1\}^n$ uma função booleana de n para n bits e $c \in \{0, 1\}^n$.

Diz-se que f satisfaz a *propriedade de Simon* se

$$f(x_1) = f(x_2) \iff x_2 = x_1 \oplus c,$$

em que a operação \oplus é a XOR (ou seja, adição módulo 2) realizada bit a bit nos dois vetores de bits.

Exemplo 6.9. A função booleana $f: \{0, 1\}^2 \rightarrow \{0, 1\}^2$ definida pela ta-

⁴Isto é, (1) se não repete valores para diferentes entradas e (2) se o seu resultado varre todas as opções de palavras de n bits. O item (1) significa que f é *injetiva* e (2) significa que f é *sobrejetiva*. Em símbolos, essas propriedades ficam:

(1) $x_1, x_2 \in \mathbb{B}_n, x_1 \neq x_2 \implies f(x_1) \neq f(x_2)$

(2) $\forall y \in \mathbb{B}_n, \exists x \in \mathbb{B}_n: y = f(x)$.

bela abaixo é 1-para-1.

x	$f(x)$
00	10
01	11
10	00
11	01

Exemplo 6.10. A função booleana $f: \{0,1\}^2 \rightarrow \{0,1\}^2$ definida pela tabela abaixo é 2-para-1.

x	$f(x)$
00	10
01	01
10	10
11	01

Essa função também satisfaz a propriedade de Simon com $c = 10$. De fato,

$$\begin{aligned} 10 &= 00 \oplus 10 & f(00) &= f(10) = 10 \\ 11 &= 01 \oplus 10 & f(01) &= f(11) = 01 . \end{aligned}$$

Observação 6.11. Nem toda função booleana 2-para-1 satisfaz a propriedade de Simon. De fato, a função $f: \{0,1\}^3 \rightarrow \{0,1\}^3$ dada por⁵

x	$f(x)$	x	$f(x)$
000	2	100	2
001	5	101	3
010	1	110	1
011	5	111	3

Essa função é 2-para-1. Se satisfizesse a propriedade de Simon, existiria c satisfazendo $f(x \oplus c) = f(x)$ para todo x . No entanto,

$$f(000) = f(100), 100 = 000 \oplus 100 \implies c = 100$$

e tem-se que

$$101 = 001 \oplus 100 \text{ mas } f(101) \neq f(001) .$$

Essa contradição significa que a propriedade de Simon não é satisfeita.

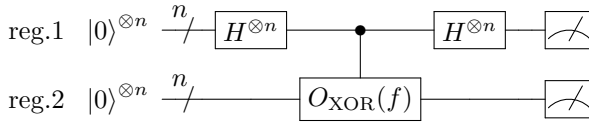
Observação 6.12. Se f satisfaz a propriedade de Simon com $c = 0 \dots 0$, então f é 1-para-1. E se f satisfaz a propriedade de Simon com $c \neq 0 \dots 0$, então f é 2-para-1.

De posse dessas definições, o problema de Simon tem o seguinte enunciado.

⁵Para simplificar, usou-se a notação que confunde uma palavra de bits com sua representação por número inteiro sem sinal. Ver observação 6.6.

Circuito

Notação compacta:



Notação expandida:

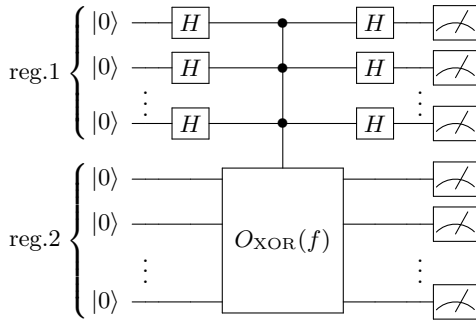


Figura 6.20: Uma iteração do algoritmo de Simon.

Contas auxiliares

Definição 6.13. Usa-se a seguinte notação, com o intuito de simplificar algumas expressões:

$$\begin{aligned}\tilde{0} &= + \\ \tilde{1} &= - .\end{aligned}$$

Dado um vetor de bits $x \in \mathbb{B}_n$, escreve-se $|\tilde{x}\rangle = |\tilde{x}_0\tilde{x}_1\ldots\tilde{x}_{n-1}\rangle$ para designar um produto tensorial de estados $|+\rangle$ e $|-\rangle$. Por exemplo,

$$|x\rangle = |0110\rangle \iff |\tilde{x}\rangle = |+-+ \rangle .$$

Definição 6.14. \mathbb{B}_n é o conjunto de todos os vetores de n bits.

Proposição 6.15. Vale que

$$H^{\otimes n} = \sum_{y \in \mathbb{B}_n} |\tilde{y}\rangle\langle y| .$$

Demonstração. Prova-se por indução em n .

Vale para $n = 1$ qubit, já que H pode ser escrito como

$$H = |+\rangle\langle 0| + |-\rangle\langle 1| \ .$$

Vale para $n = 2$ qubits, pois

$$\begin{aligned} H^{\otimes 2} &= H \otimes H \\ &= (|+\rangle\langle 0| + |-\rangle\langle 1|) \otimes (|+\rangle\langle 0| + |-\rangle\langle 1|) \\ &= |++\rangle\langle 00| + |+-\rangle\langle 01| + |-+\rangle\langle 10| + |--\rangle\langle 11| \\ &= \sum_{y \in \mathbb{B}_2} |\tilde{y}\rangle\langle y| \ . \end{aligned}$$

Supõe-se, então, que seja válido para n qubits. Verifica-se o caso $n + 1$:

$$\begin{aligned} H^{\otimes n+1} &= H^{\otimes n} \otimes H \\ &= \left(\sum_{y \in \mathbb{B}_n} |\tilde{y}\rangle\langle y| \right) \otimes (|+\rangle\langle 0| + |-\rangle\langle 1|) \\ &= \left(\sum_{y \in \mathbb{B}_n} |\tilde{y}_0 \tilde{y}_1 \dots \tilde{y}_{n-1}\rangle\langle y_0 y_1 \dots y_{n-1}| \right) \otimes \left(\underbrace{|+\rangle\langle 0|}_{y_n=0} + \underbrace{|-\rangle\langle 1|}_{y_n=1} \right) \\ &= \sum_{y \in \mathbb{B}_{n+1}} |\tilde{y}_0 \tilde{y}_1 \dots \tilde{y}_{n-1} \tilde{y}_n\rangle\langle y_0 y_1 \dots y_{n-1} y_n| \\ &= \sum_{y \in \mathbb{B}_{n+1}} |\tilde{y}\rangle\langle y| \ . \end{aligned}$$

Isso conclui a indução em n . □

Proposição 6.16. *Vale que*

$$|\tilde{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle \ ,$$

em que $x \cdot y = x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1}$.

Demonstração. Mostra-se por indução em n .

Para $n = 1$ qubit, tem-se que

$$\begin{aligned} |\tilde{0}\rangle &= |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ |\tilde{1}\rangle &= |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \ . \end{aligned}$$

Para $n = 2$ qubits, tem-se que

$$\begin{aligned} |\tilde{0}\tilde{0}\rangle &= |++\rangle = \frac{1}{\sqrt{2^2}} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ |\tilde{0}\tilde{1}\rangle &= |+-\rangle = \frac{1}{\sqrt{2^2}} (|00\rangle - |01\rangle + |10\rangle - |11\rangle) \\ |\tilde{1}\tilde{0}\rangle &= |-+\rangle = \frac{1}{\sqrt{2^2}} (|00\rangle + |01\rangle - |10\rangle - |11\rangle) \\ |\tilde{1}\tilde{1}\rangle &= |--\rangle = \frac{1}{\sqrt{2^2}} (|00\rangle - |01\rangle - |10\rangle + |11\rangle) , \end{aligned}$$

que pode ser resumido em

$$|\tilde{x}\rangle = \frac{1}{\sqrt{2^2}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle .$$

Assume-se que o enunciado seja válido para n qubits. O caso $n+1$ fica como a seguir.

$$|\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1} \tilde{x}_n\rangle = |\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1}\rangle \otimes |\tilde{x}_n\rangle$$

Caso $\tilde{x}_n = +$, tem-se

$$\begin{aligned} & |\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1} +\rangle \\ &= |\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1}\rangle \otimes |+\rangle \\ &= \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} |y\rangle \right) \otimes |+\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} |y_0 \dots y_{n-1}\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} \left(\underbrace{|y_0 \dots y_{n-1} 0\rangle}_{y_n=0} + \underbrace{|y_0 \dots y_{n-1} 1\rangle}_{y_n=1} \right) \\ &= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y + 0 \cdot 0} \underbrace{|y_0 \dots y_{n-1} 0\rangle}_{y_n=0} + \right. \\ &\quad \left. + (-1)^{(x_0 \dots x_{n-1}) \cdot y + 0 \cdot 1} \underbrace{|y_0 \dots y_{n-1} 1\rangle}_{y_n=1} \right) \\ &= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \mathbb{B}_{n+1}} (-1)^{(x_0 \dots x_{n-1} 0) \cdot y} |y_0 \dots y_{n-1} y_n\rangle . \end{aligned}$$

No caso $\tilde{x}_n = -$, tem-se

$$\begin{aligned}
& |\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1} - \rangle \\
&= |\tilde{x}_0 \tilde{x}_1 \dots \tilde{x}_{n-1} \rangle \otimes |- \rangle \\
&= \left(\frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} |y\rangle \right) \otimes |- \rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} |y_0 \dots y_{n-1}\rangle \otimes \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y} \left(\underbrace{|y_0 \dots y_{n-1} 0\rangle}_{y_n=0} - \underbrace{|y_0 \dots y_{n-1} 1\rangle}_{y_n=1} \right) \\
&= \frac{1}{\sqrt{2^{n+1}}} \left(\sum_{y \in \mathbb{B}_n} (-1)^{(x_0 \dots x_{n-1}) \cdot y + 1 \cdot 0} \underbrace{|y_0 \dots y_{n-1} 0\rangle}_{y_n=0} + \right. \\
&\quad \left. + (-1)^{(x_0 \dots x_{n-1}) \cdot y + 1 \cdot 1} \underbrace{|y_0 \dots y_{n-1} 1\rangle}_{y_n=1} \right) \\
&= \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \mathbb{B}_{n+1}} (-1)^{(x_0 \dots x_{n-1} 1) \cdot y} |y_0 \dots y_{n-1} y_n\rangle .
\end{aligned}$$

Isso conclui a demonstração. \square

Proposição 6.17. *O produto tensorial de n operadores de Hadamard é dado por*

$$H^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x, y \in \mathbb{B}_n} (-1)^{x \cdot y} |x\rangle \langle y| .$$

Demonstração. Usando as proposições 6.15 e 6.16, tem-se que

$$H^{\otimes n} = \sum_{y \in \mathbb{B}_n} |\tilde{y}\rangle \langle y| = \sum_{y \in \mathbb{B}_n} \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} (-1)^{x \cdot y} |x\rangle \langle y| . \quad \square$$

Proposição 6.18. *A aplicação de $H^{\otimes n}$ a um estado $|x\rangle = |x_0 x_1 \dots x_{n-1}\rangle$ na base computacional é dada por*

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle ,$$

em que $x \cdot y = x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1}$.

Demonstração. Usando as proposição 6.17, tem-se que

$$\begin{aligned} H^{\otimes n} |x\rangle &= \frac{1}{\sqrt{2^n}} \sum_{y,z \in \mathbb{B}_n} (-1)^{y \cdot z} |y\rangle \langle z| |x\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{y,z \in \mathbb{B}_n} (-1)^{y \cdot z} |y\rangle \delta_{z,x} \\ &= \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{y \cdot x} |y\rangle . \end{aligned}$$

□

Observação 6.19. A soma e o produto em

$$x \cdot y = x_0 y_0 + x_1 y_1 + \dots + x_{n-1} y_{n-1} \quad (\text{int})$$

podem ser entendidos como operações com números ou como operações com bits

$$x \cdot y = x_0 y_0 \oplus x_1 y_1 \oplus \dots \oplus x_{n-1} y_{n-1} \quad (\text{bit})$$

em que o produto é dado pela AND e a soma \oplus é dada pela XOR. Ambas as expressões resultam no mesmo sinal $(-1)^{x \cdot y}$, pois a expressão (bit) corresponde a (int) módulo 2, visto que a AND se comporta como um produto e a XOR, como uma adição módulo 2.

Etapas da subrotina de Simon

As etapas da subrotina quântica são mostradas em detalhes no texto que segue. Inicialmente, aplica-se $H^{\otimes n}$ ao primeiro registrador, obtendo-se

$$|\psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} |x\rangle |0\rangle ,$$

em que o primeiro ket engloba n qubits e representa o primeiro registrador, e o segundo ket contém n bits e representa o segundo registrador.

A aplicação do oráculo na etapa 2 mantém o primeiro registrador e faz a XOR bit a bit de 0 com $f(x)$:

$$\begin{aligned} |\psi_2\rangle &= O_{\text{XOR}} |\psi_1\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} O_{\text{XOR}} |x\rangle |0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} |x\rangle |f(x)\rangle . \end{aligned}$$

Na etapa 3, aplica-se novamente $H^{\otimes n}$ ao primeiro registrador:

$$\begin{aligned}
 |\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} (H^{\otimes n} |x\rangle) |f(x)\rangle \\
 &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}_n} \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle |f(x)\rangle \\
 &= \frac{1}{2^n} \sum_{x \in \mathbb{B}_n} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle |f(x)\rangle .
 \end{aligned} \tag{*}$$

A medida na base computacional, na etapa 4, faz o estado do sistema colapsar em $|y\rangle |z\rangle$, com $z = f(x)$. Como f tem a propriedade de Simon, os únicos valores que resultam em z pela aplicação de f são

$$z = f(x_1) = f(x_2) , \quad x_2 = x_1 \oplus c .$$

Probabilidades nas medições

Caso $c \neq 0$, o coeficiente multiplicando o estado $|y\rangle |z\rangle$ em (*) é dado por

$$\begin{aligned}
 a_{y,z} &= \frac{1}{2^n} ((-1)^{x_1 \cdot y} + (-1)^{x_2 \cdot y}) \\
 &= \frac{1}{2^n} ((-1)^{x_1 \cdot y} + (-1)^{(x_1 \oplus c) \cdot y}) \\
 &= \frac{1}{2^n} ((-1)^{x_1 \cdot y} + (-1)^{(x_1 \cdot y) \oplus (c \cdot y)}) \\
 &= \frac{1}{2^n} ((-1)^{x_1 \cdot y} + (-1)^{x_1 \cdot y} (-1)^{c \cdot y}) \\
 &= \frac{1}{2^n} (-1)^{(x_1 \cdot y)} (1 + (-1)^{c \cdot y}) \\
 &= \begin{cases} 0 & c \cdot y = 1 , \\ (-1)^{(x_1 \cdot y)} \frac{2}{2^n} & c \cdot y = 0 . \end{cases}
 \end{aligned}$$

A probabilidade de se encontrar o sistema no estado $|y\rangle |z\rangle$ é, então,

$$p_{y,z} = |a_{y,z}|^2 = \begin{cases} 0 & c \cdot y = 1 \\ \frac{2^2}{2^{2n}} & c \cdot y = 0 \end{cases} \tag{p1}$$

Assim, a medida só fornece vetores de bits y perpendiculares a c . A informação que se ganha para encontrar c é a equação

$$c \cdot y = c_1 y_1 \oplus c_2 y_2 \oplus \dots \oplus c_n y_n = 0 .$$

Caso $c = 0$, o coeficiente em $(*)$ fica apenas $a_{y,z} = \frac{1}{2^n}(-1)^{x_1 \cdot y}$ e a probabilidade de se encontrar o sistema em $|y\rangle|z\rangle$ é

$$p_{y,z} = |a_{y,z}|^2 = 2^{-2n} . \quad (\text{p2})$$

Logo, qualquer vetor de bits y pode sair como resultado da medição.

Encontrando o valor do período c – exemplo

Primeiramente, apresenta-se o processamento para encontrar o período c em um exemplo, com o objetivo de facilitar a compreensão do método no caso geral.

Exemplo 6.20 (Encontrando período c com o algoritmo de Simon).

Seja $n = 4$ bits. Aplica-se a primeira iteração da subrotina quântica do algoritmo de Simon. Suponha que se obteve o resultado $y^{(1)} = 0111$. Esse resultado gera a equação

$$y^{(1)} \cdot c = 0 \implies c_2 \oplus c_3 \oplus c_4 = 0 \implies c_2 = c_3 \oplus c_4 .$$

Continua-se aplicando a subrotina até se obter $n - 1 = 3$ equações LI.

A segunda iteração fornece $y^{(2)} = 1001$. Esse resultado corresponde à equação

$$y^{(2)} \cdot c = 0 \implies c_1 \oplus c_4 = 0 ,$$

e o sistema, após simplificação, fica

$$\begin{cases} c_2 = c_3 \oplus c_4 \\ c_1 = c_4 . \end{cases}$$

Como ainda não são 3 equações LI, continua-se a iteração.

Na terceira iteração, o resultado é $y^{(3)} = 1110$. A equação correspondente é

$$y^{(3)} \cdot c = 0 \implies c_1 \oplus c_2 \oplus c_3 = 0 \implies c_4 \oplus (c_3 \oplus c_4) \oplus c_3 = 0 \implies 0 = 0 ,$$

e essa equação não fornece informação útil. O sistema continua sendo de 2 equações LI:

$$\begin{cases} c_2 = c_3 \oplus c_4 \\ c_1 = c_4 . \end{cases}$$

Na quarta iteração obtém-se $y^{(4)} = 0001$, e a equação que esse resultado gera é

$$y^{(4)} \cdot c = 0 \implies c_4 = 0 .$$

O sistema fica

$$\begin{cases} c_2 = c_3 \\ c_1 = 0 \\ c_4 = 0 \end{cases}.$$

Agora são $3 = n - 1$ equações LI. As soluções são $c' = 0000$ e $c'' = 0110$. Poder-se-ia concluir que f é 2-para-1 com período $c = 0110$. No entanto, a probabilidade de se estar errado nesse caso é a probabilidade de se obter 4 resultados no mesmo subespaço de dimensão 3, sendo que f seria 1-para-1 e os $2^n = 2^4$ resultados seriam equiprováveis:

$$\varepsilon_4 \lesssim 1 \cdot 1 \cdot 1 \cdot \frac{2^3}{2^4} = \frac{1}{2}$$

Para reduzir a probabilidade de erro, aplica-se a subrotina novamente. Suponha que obtém-se $y^{(5)} = 1111$. Verifica-se se $y^{(5)} \perp c''$ ou não. Caso não fosse, concluir-se-ia que haveria mais uma equação independente e o sistema só teria solução $c' = 0$. Não é esse o caso aqui, pois

$$y^{(5)} \cdot c'' = 1111 \cdot 0110 = 0 \oplus 1 \oplus 1 \oplus 0 = 0 \implies y^{(5)} \perp c''.$$

Poderia concluir-se, nessa etapa, que f é 2-para-1 com probabilidade de erro igual à probabilidade de se sortear aleatoriamente 5 vetores e todos caírem no mesmo subespaço vetorial de dimensão 3:

$$\varepsilon_5 \lesssim 1 \cdot 1 \cdot 1 \cdot \frac{2^3}{2^4} \cdot \frac{2^3}{2^4} = \frac{1}{4}.$$

Terminando o algoritmo na iteração 5, tem-se que f é 2-para-1 e que o período é $c = 0110$.

A título de curiosidade, a f utilizada neste exemplo é disposta na tabela abaixo, em que A, B, C, D, E, F, G e H denotam 8 palavras distintas de 4 bits.

x	$f(x)$	x	$f(x)$
0000	A	1000	E
0001	B	1001	F
0010	C	1010	G
0011	D	1011	H
0100	C	1100	G
0101	D	1101	H
0110	A	1110	E
0111	B	1111	F

Repare que, de fato, f é 2-para-1 com $c = 0110$.

Encontrando o valor do período c – caso geral

Repetindo a subrotina m vezes, obtém-se os resultados $y^{(1)}, \dots, y^{(m)}$ das medidas no registrador 1 e o sistema de equações lineares na incógnita $c = c_1 c_2 \dots c_n$:

$$\begin{cases} y^{(1)} \cdot c = 0 \\ y^{(2)} \cdot c = 0 \\ \vdots \\ y^{(m)} \cdot c = 0 \end{cases} \quad (s)$$

Esse sistema sempre admite a solução $c = 0$. Supõe-se que se tenha obtido, após a aplicação da subrotina por um número suficiente de vezes, um sistema linear com um número suficiente de equações linearmente independentes (ficará mais claro o que significaria “suficiente” nesse contexto).

Observação 6.21. Para analisar esse sistema, é interessante considerar \mathbb{B}_n como espaço vetorial sobre os escalares $\mathbb{B} = \{0, 1\}$ e com a soma de vetores dada pela XOR bit a bit \oplus . É possível verificar que esse espaço satisfaz os axiomas de espaço vetorial. Além disso, é possível imitar o produto interno em \mathbb{R}^n ou \mathbb{C}^n com a operação $r \cdot s = r_1 s_1 \oplus \dots \oplus r_n s_n$.

O espaço \mathbb{B}_n tem dimensão n e contém 2^n vetores. A maioria dos resultados de Álgebra Linear se mantém para esse caso, exceto que esses espaços vetoriais têm um número finito de vetores e que é possível ter $x \neq 0$ e $x \cdot x = 0$, de forma que o produto \cdot não é um produto interno em \mathbb{B}_n . Os subespaços de \mathbb{B}_n têm dimensão $2^m, m \leq n$. O subespaço gerado por $c \neq 0$ é $\{0, c\}$ e tem dimensão 1. Se W é um subespaço, então o conjunto W^\perp dos vetores perpendiculares a W é também um subespaço, e vale que $\dim W + \dim W^\perp = \dim \mathbb{B}_n = n$.

Os livros de Códigos Corretores de Erros (da Computação Clássica) costumam denotar \mathbb{B} por $GF(2)$, chamado *campo de Galois* (*Galois field*) de dois elementos. Uma referência contendo um resumo da teoria relacionada ao espaço vetorial $GF(2)^n$ é [13], seção 2.4, p.75-80.

Se f for 1-para-1, espera-se que o sistema admita apenas a solução trivial $c = 0$. Os valores $y^{(1)}, \dots, y^{(m)}$ podem ser quaisquer dos 2^n vetores de bits em \mathbb{B}_n , por causa da equação (p2). Como a dimensão de \mathbb{B}_n é n , há no máximo n vetores de bits LI nesse espaço. Isso significa que o sistema acima é equivalente a um sistema de n equações LI, e que só admite a solução trivial $c = 0$, como esperado.

Por outro lado, se f for 2-para-1, espera-se que o sistema tenha duas soluções: 0 e $c \neq 0$. Nesse caso, os valores $y^{(1)}, \dots, y^{(m)}$ são, obrigatoriamente, perpendiculares ao vetor c . Se W é o subespaço gerado por c , tem-se que $y^{(1)}, \dots, y^{(m)} \in W^\perp$ e que $\dim W = 1$ e $\dim W^\perp = n - 1$, de

forma que $\dim W + \dim W^\perp = \dim \mathbb{B}_n$. Assim, há no máximo $n - 1$ vetores LI e perpendiculares a c . O sistema (s) é equivalente a um sistema de $n - 1$ equações LI, e apresenta uma variável livre c_j , que pode assumir os valores 0 ou 1, gerando as soluções 0 e $c \neq 0$, como era esperado.

Resumindo, se as m repetições da subrotina quântica do algoritmo de Simon produzirem um sistema de equações com o máximo possível de equações LI, a solução do sistema fornecerá c e, dependendo se há apenas a solução nula ou se, além dessa, há uma solução $c \neq 0$, pode-se distinguir os casos “ f é 1-para-1” ou “ f é 2-para-1”. O número de repetições m requerido é proporcional a n .

Probabilidade de erro

Em relação à probabilidade de erro no caso geral, tem-se o seguinte. A partir de um número de iterações suficientemente grande (da ordem de n), a cada nova iteração, ou se descobre que f é 1-para-1 (resultado cada vez mais improvável) ou se escolhe que f é 2-para-1 com probabilidade de erro:

$$\varepsilon_m \lesssim \underbrace{1 \dots 1}_{n-1 \text{ vezes}} \cdot \underbrace{\frac{2^{n-1}}{2^n} \dots \frac{2^{n-1}}{2^n}}_{m-(n-1) \text{ vezes}} = \frac{1}{2^{m-n+1}}.$$

Essa estimativa não é exata, pois está considerando o caso em que $n - 1$ resultados forneceram vetores de bits não-nulos e LI, e os outros resultados caíram no subespaço gerado pelos $n - 1$ vetores LI. Poderia ter acontecido de se obter menos vetores LI e os outros caírem no subespaço gerado por eles, apesar de parecer menos provável. Essa estimativa, contudo, serve para dar uma pista quanto à quantidade de iterações do algoritmo quântico. Dessa forma, se o número de iterações m for da ordem de n , a probabilidade de erro pode ser feita menor que $1/2$.

6.5.3 Algoritmo Clássico

Algoritmo Clássico Determinístico

Um algoritmo clássico para o Problema de Simon consiste em escolher entradas x em \mathbb{B}_n , calcular $f(x)$ e comparar com os outros valores já obtidos, até que se encontre um par de vetores distintos $x_{(1)}$ e $x_{(2)}$ com $f(x_{(1)}) = f(x_{(2)})$ ou até que se possa concluir que f é 1-para-1.

Supondo que seja possível armazenar todas as entradas testadas e seus resultados pela aplicação da f , na pior das hipóteses, deve-se calcular f para metade das entradas mais uma, isto é, $2^n/2 + 1$ vezes. Caso f seja 1-para-1, todos os resultados seriam distintos, e caso f seja 2-para-1, na

pior das hipóteses, na tentativa de número $2^n/2 + 1$ será obtido um valor repetido após aplicação da função.

Algoritmo Clássico Probabilístico

Para melhorar o desempenho do algoritmo determinístico acima, pode-se relaxar o desempenho, permitindo-se uma probabilidade de erro ε na escolha. Sorteia-se aleatoriamente x dentre o conjunto de entradas não testadas ainda, calcula-se $f(x)$ e compara-se o valor obtido com o fornecido pelas entradas já testadas. Repete-se por um número suficiente de tentativas ou até algum par ser encontrado; se nenhum par foi encontrado, decide-se por “ f é 1-para-1” e se for encontrado, decide-se por “ f é 2-para-1” e utiliza-se o par $x_{(1)}, x_{(2)}$ encontrado para calcular $c = x_{(1)} \oplus x_{(2)}$.

Após m iterações, o número de pares já testados é N_{ob} , em que

$$N_{\text{ob}} = \binom{m}{2} = \frac{m(m-1)}{2} ,$$

isto é, o número de pares que se pode formar dentre m elementos distintos sem importar a ordem em que se encontram no par.

Caso f seja 2-para-1, o número de pares desejado (ou seja, que resultam no mesmo valor após aplicação de f) é N_{des} dado por

$$N_{\text{des}} = \frac{2^n}{2} .$$

A probabilidade de pelo menos um par desejado ter sido obtido após k iterações é

$$p_m = \frac{N_{\text{ob}}}{N_{\text{des}}} = \frac{m(m-1)}{2^n} .$$

Caso nenhum par desejado tenha sido encontrado, opta-se por “ f é 1-para-1” com probabilidade de erro dada por

$$\varepsilon_m = 1 - p_m = 1 - \frac{m(m-1)}{2^n} .$$

Para que a probabilidade de erro seja $\varepsilon < 1/2$, deve-se ter

$$\varepsilon_m < \frac{1}{2} \implies \frac{m(m-1)}{2^n} > \frac{1}{2} \implies m^2 - m - 2^{n-1} > 0 .$$

Resolvendo para $m > 0$, deve-se ter

$$m > \frac{1 + \sqrt{1 + 2^{n+1}}}{2} ,$$

logo m deve ser da ordem de $2^{n/2}$ iterações.

6.5.4 Comparação de Desempenho

O desempenho dos algoritmos clássico determinístico, clássico probabilístico e quântico são resumidos na tabela abaixo.

Algoritmo	Desempenho (# aplicações de f)
Class. Det.	da ordem de $2^n/2$ aplicações
Class. Prob.	da ordem de $2^{n/2}$ aplicações
Quântico	da ordem de n aplicações

Tabela 6.5: Comparação de desempenho entre os algoritmos quântico, clássico determinístico e clássico probabilístico (com probabilidade de erro $< 50\%$) para o Problema de Simon.

Da mesma forma como no problema de Deutsch-Jozsa, essa comparação tem limitações, mas serve como laboratório para testar em que situações a Computação Quântica pode trazer vantagem computacional em relação à Computação Clássica. Em particular, esse é um exemplo em que o algoritmo quântico apresenta ganho exponencial em desempenho em relação aos algoritmos clássicos existentes.

6.6 Algoritmo de Busca de Grover

Esta seção aborda o Algoritmo de Grover, um algoritmo quântico que permite encontrar um elemento em uma lista não ordenada. Uma descrição mais rigorosa desse problema, bem como as alternativas clássicas para o mesmo são apresentadas neste texto. As referências utilizadas nesta parte são o livro [17], capítulo 3, e as videoaulas U2.6 da subunidade SU4 do curso [19].

6.6.1 Problema de Grover

O problema de Grover consiste em encontrar um elemento específico em uma lista de 2^n itens. A lista é formada por palavras binárias de n dígitos. Há uma função booleana $f: \{0, 1\}^n \rightarrow \{0, 1\}$ que só sinaliza '1' para uma entrada x_0 em particular, que pode ser desconhecida. Em outras palavras, a função f pode ser descrita por

$$f(x) = \begin{cases} 0, & x \neq x_0 \\ 1, & x = x_0 \end{cases}.$$

A maneira de saber se o item x considerado corresponde ao desejado é por meio da aplicação de f . Essa função poderia ser chamada de “teste”, e

caso se queira saber se o item x da lista é o desejado (em outras palavras, se $x = x_0$), deve-se fazer o teste em x e ver se o teste resulta em '1' (item desejado foi encontrado) ou '0' (o item testado não é o desejado).

Problema de Grover. *Encontrar a única entrada $x_0 \in \{0, 1\}^n$ tal que o resultado do teste f sinaliza '1'. Ou seja, encontrar único x_0 com*

$$f(x_0) = 1 .$$

Em seguida, um algoritmo quântico é apresentado para resolver o problema de forma mais eficiente do que seria possível classicamente.

6.6.2 Algoritmo de Grover

O algoritmo de Grover é um algoritmo quântico para resolução do problema de Grover, que consiste em encontrar o elemento em uma lista por meio de um teste f , e o elemento desejado é o único x_0 que faz com que o teste sinalize 1. O algoritmo quântico consiste em aplicar uma subrotina quântica, o operador de Grover, por um número de vezes da ordem de \sqrt{N} , em que $N = 2^n$ é o número de itens na lista, indexados por n bits.

Algoritmo de Grover

Entrada: $O_F(f) = O$ (oráculo de fase associado à função booleana f)

Procedimento:

- etapa 0: $|0\rangle^{\otimes n}$ preparação do estado inicial
- etapa 1: $H^{\otimes n} |0\rangle^{\otimes n}$ superposição dos estados na base computacional
- etapa 2: G aplicação do operador de Grover
- etapa 3: repetir etapa 2 por k vezes, com

$$k = \frac{\arccos\left(\frac{1}{\sqrt{N}}\right)}{\arccos\left(\frac{N-2}{N}\right)} , \quad N = 2^n .$$

Operador de Grover G :

- etapa 1: $O_F |+\rangle^{\otimes n}$ aplicação de f (oráculo de fase)
- etapa 2: $H^{\otimes n}$
- etapa 3: $2|0\rangle\langle 0| - I$
- etapa 4: $H^{\otimes n}$

Saída: Leitura do registrador fornece o item x_0 buscado, com probabilidade de acerto

$$P_a > \frac{N-1}{N}, \quad N = 2^n .$$

Circuito

Notação compacta:

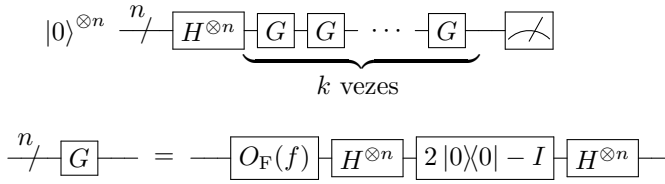


Figura 6.21: Algoritmo de Busca de Grover.

Notação auxiliar

Para facilitar, lista-se abaixo a notação utilizada nesse algoritmo:

$$|0\rangle = |0 \dots 0\rangle = |0\rangle^{\otimes n}$$

$$|\psi\rangle := |+\rangle^{\otimes n}$$

\mathbb{B}_n : conjunto de todas as palavras de n bits

$$N := 2^n \quad \begin{cases} n: \text{número de qubits de cada item da lista} \\ N: \text{número de itens na lista} \end{cases}$$

$$|\alpha\rangle := \sum_{\substack{x \in \mathbb{B}_n \\ x \neq x_0}} \frac{|x\rangle}{\sqrt{N-1}}$$

$$|\beta\rangle := |x_0\rangle : \text{item desejado na lista}$$

$S := \text{span}_{\mathbb{R}}\{|\alpha\rangle, |\beta\rangle\}$: espaço vetorial gerado por $|\alpha\rangle, |\beta\rangle$ com escalares reais

Contas auxiliares

Vale que:

$$\begin{aligned} |\psi\rangle &= |+\rangle^{\otimes n} \\ &= \sum_{x \in \mathbb{B}_n} \frac{|x\rangle}{\sqrt{N}} \\ &= \frac{\sqrt{N-1}}{\sqrt{N}} \sum_{\substack{x \in \mathbb{B}_n \\ x \neq x_0}} \frac{|x\rangle}{\sqrt{N-1}} + \frac{|x_0\rangle}{\sqrt{N}} \quad (\psi) \\ &= \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle + \frac{1}{\sqrt{N}} |\beta\rangle \end{aligned}$$

A aplicação de O_F em $|\alpha\rangle$ e $|\beta\rangle$ fica:

$$\begin{aligned}
 O_F |\alpha\rangle &= O_F \sum_{\substack{x \in \mathbb{B}_n \\ x \neq x_0}} \frac{|x\rangle}{\sqrt{N-1}} \\
 &= \sum_{\substack{x \in \mathbb{B}_n \\ x \neq x_0}} \frac{1}{\sqrt{N-1}} O_F |x\rangle \\
 &= \sum_{\substack{x \in \mathbb{B}_n \\ x \neq x_0}} \frac{1}{\sqrt{N-1}} |x\rangle \\
 &= |\alpha\rangle ,
 \end{aligned} \tag{\alpha}$$

$$\begin{aligned}
 O_F |\beta\rangle &= O_F |x_0\rangle \\
 &= -|x_0\rangle \\
 &= -|\beta\rangle .
 \end{aligned} \tag{\beta}$$

O operador G pode ser escrito como:

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

em que foi definido $|\psi\rangle = |+\rangle^{\otimes n}$.

Primeira aplicação do operador G

Antes de aplicar o operador G pela primeira vez, prepara-se o estado inicial fazendo uma superposição com pesos iguais em cada qubit:

$$\begin{aligned}
 |\psi_0\rangle &= H^{\otimes n} |0\rangle^{\otimes n} \\
 &= |+\rangle^{\otimes n} \\
 &= |\psi\rangle \\
 &= \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle + \frac{1}{\sqrt{N}} |\beta\rangle \quad \text{usando } (\psi) .
 \end{aligned}$$

Nota-se que $|\psi_0\rangle$ pertence ao subespaço vetorial S gerado por $|\alpha\rangle, |\beta\rangle$ e com escalares reais. Isso permitirá uma interpretação geométrica muito útil para o entendimento do algoritmo.

Na etapa 1 de G , aplica-se o oráculo de fase ao estado inicial $|\psi\rangle$:

$$\begin{aligned}
 |\psi_1\rangle &= O_F |\psi_0\rangle \\
 &= O_F \left(\frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle + \frac{1}{\sqrt{N}} |\beta\rangle \right) \quad \text{usando } (|\psi\rangle) \\
 &= \frac{\sqrt{N-1}}{\sqrt{N}} O_F |\alpha\rangle + \frac{1}{\sqrt{N}} O_F |\beta\rangle \\
 &= \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle - \frac{1}{\sqrt{N}} |\beta\rangle \quad \text{usando } (\alpha) \text{ e } (\beta)
 \end{aligned}$$

O vetor $|\psi_1\rangle$ continua no espaço S , e a aplicação do oráculo de fase pode ser visualizada como uma reflexão em torno do eixo $|\alpha\rangle$.

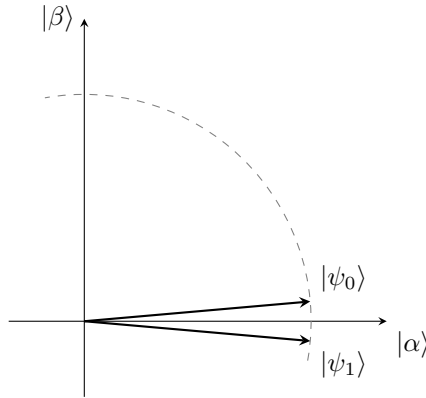


Figura 6.22: Aplicação do oráculo de fase equivale a uma reflexão em relação ao eixo $|\alpha\rangle$.

As etapas 2, 3 e 4 de G redundam em aplicar $2|\psi\rangle\langle\psi| - I$ em $|\psi_1\rangle$, conforme equação (G). E a aplicação desse operador pode ser vista geometricamente de acordo com a figura abaixo.

$$\begin{aligned}
 |\psi_2\rangle &= (2|\psi\rangle\langle\psi| - I) |\psi_1\rangle \\
 &= 2|\psi\rangle\langle\psi| |\psi_1\rangle - |\psi_1\rangle
 \end{aligned}$$

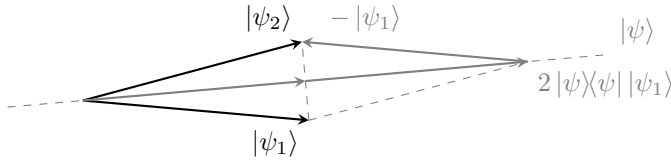


Figura 6.23: Aplicação do operador $2|\psi\rangle\langle\psi| - I$ equivale a uma reflexão em relação à reta determinada pelo vetor $|\psi\rangle$.

Dessa forma, a primeira aplicação do operador G fornece o seguinte.

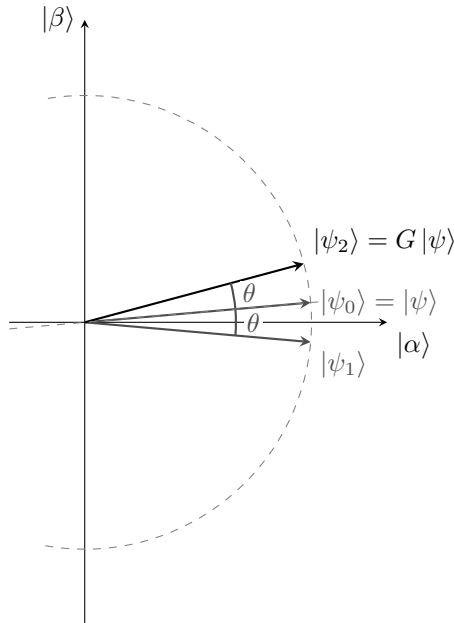


Figura 6.24: Aplicação do operador G . O efeito corresponde à rotação do vetor por um ângulo θ no sentido anti-horário.

Aplicações subsequentes do operador G

As aplicações sucessivas do operador de Grover têm o mesmo efeito descrito anteriormente. Cada aplicação de G corresponde a uma rotação no sentido anti-horário. A figura a seguir ilustra a l -ésima aplicação de G .

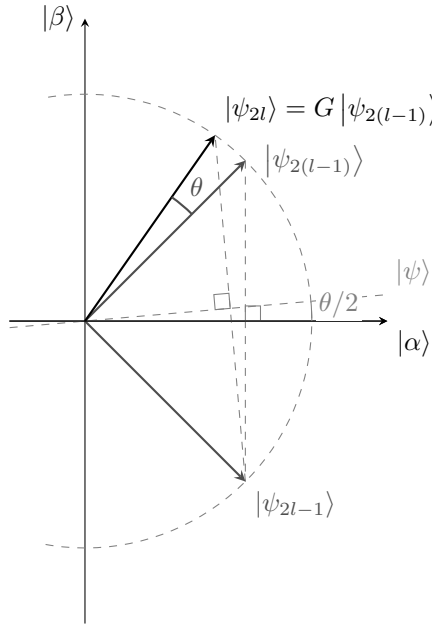


Figura 6.25: Aplicação l -ésima do operador G . O efeito continua correspondendo à rotação do vetor por um ângulo θ no sentido anti-horário.

O operador G é aplicado por k vezes até que o vetor resultante esteja o mais próximo possível do eixo $|\beta\rangle$.

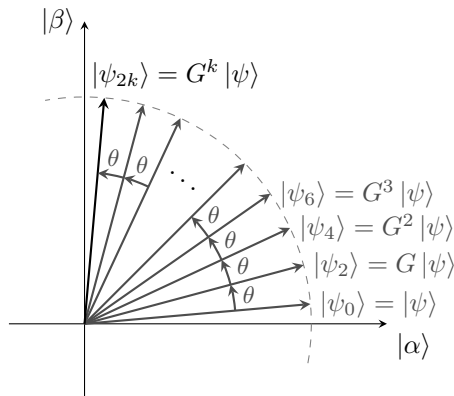


Figura 6.26: Aplicações sucessivas do operador G .

No livro [17], seção 3.3, demonstra-se algebricamente que uma aplicação de G produz uma rotação em sentido horário de um mesmo ângulo θ .

Número necessário de aplicações de G

O número de aplicações k necessário é dado por

$$k\theta + \frac{\theta}{2} = \frac{\pi}{2} \implies k = \frac{\pi - \theta}{2\theta},$$

em que se arredonda k para o inteiro mais próximo.

O ângulo θ (em radianos) é o ângulo que $G|\psi\rangle$ faz com $|\psi\rangle$. Pode ser obtido por (ver figura 6.24):

$$\begin{aligned} \cos \frac{\theta}{2} &= \langle \beta | \psi \rangle = \frac{\sqrt{N-1}}{\sqrt{N}} \quad \text{usando } (\psi) \\ \sin \frac{\theta}{2} &= \sqrt{1 - |\langle \beta | \psi \rangle|^2} = \frac{1}{\sqrt{N}} \end{aligned}$$

Fazendo-se a seguinte manipulação algébrica, pode-se encontrar outra expressão equivalente para o ângulo.

$$\begin{aligned} |\psi_1\rangle &= \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle - \frac{1}{\sqrt{N}} |\beta\rangle \\ &= \frac{\sqrt{N-1}}{\sqrt{N}} |\alpha\rangle + \frac{1}{\sqrt{N}} |\beta\rangle - \frac{2}{\sqrt{N}} |\beta\rangle \\ &= |\psi\rangle - \frac{2}{\sqrt{N}} |\beta\rangle \quad \text{usando } (\psi) \end{aligned}$$

$$\begin{aligned} |\psi_2\rangle &= G|\psi\rangle \\ &= (2|\psi\rangle\langle\psi| - I) |\psi_1\rangle \\ &= (2|\psi\rangle\langle\psi| - I) \left(|\psi\rangle - \frac{2}{\sqrt{N}} |\beta\rangle \right) \\ &= 2|\psi\rangle\langle\psi|\psi\rangle - |\psi\rangle - \frac{4}{\sqrt{N}} |\psi\rangle\langle\psi|\beta\rangle + \frac{2}{\sqrt{N}} |\beta\rangle \\ &= 2|\psi\rangle - |\psi\rangle - \frac{4}{\sqrt{N}} \frac{1}{\sqrt{N}} |\psi\rangle + \frac{2}{\sqrt{N}} |\beta\rangle \quad \text{usando } (\psi) \\ &= \frac{N-4}{N} |\psi\rangle + \frac{2}{\sqrt{N}} |\beta\rangle \end{aligned}$$

$$\begin{aligned}
\cos \theta &= \langle \psi | G | \psi \rangle \\
&= \langle \psi | \left(\frac{N-4}{N} |\psi\rangle + \frac{2}{\sqrt{N}} |\beta\rangle \right) \\
&= \frac{N-4}{N} \langle \psi | \psi \rangle + \frac{2}{\sqrt{N}} \langle \psi | \beta \rangle \\
&= \frac{N-4}{N} + \frac{2}{\sqrt{N}} \frac{1}{\sqrt{N}} \quad \text{usando } \langle \psi | \beta \rangle = \frac{1}{\sqrt{N}} \\
&= \frac{N-2}{N} .
\end{aligned}$$

Portanto, pode-se escrever

$$\theta = 2 \arccos\left(\frac{\sqrt{N}-1}{\sqrt{N}}\right) = 2 \arcsin\left(\frac{1}{\sqrt{N}}\right) = \arccos\left(\frac{N-2}{N}\right) ,$$

e o valor de k é dado por

$$k = \frac{\pi - \theta}{2\theta} = \frac{\pi - \arccos\left(\frac{N-2}{N}\right)}{2 \arccos\left(\frac{N-2}{N}\right)}$$

ou por

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

É possível verificar que o número k é da ordem de \sqrt{N} . De acordo com [17], p.56, tem-se

$$\lim_{N \rightarrow +\infty} k = +\infty$$

$$\lim_{N \rightarrow +\infty} \frac{k}{\sqrt{N}} = \frac{\pi}{4}$$

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

Probabilidade de acerto

Ao aplicar a subrotina G por um número k de vezes especificado anteriormente, obtém-se o estado final o mais próximo possível de $|\beta\rangle = |x_0\rangle$, o item desejado. A projeção na direção desse vetor permite encontrar a probabilidade de acerto do algoritmo. O estado final $G^k |\psi\rangle$ faz um ângulo com o estado $|x_0\rangle$ menor que $\theta/2$, pois se fosse maior ou igual, seria possível aplicar novamente o operador G para reduzi-lo. Portanto, o ângulo $\cos \theta(|x_0\rangle, G^k |\psi\rangle)$ entre o estado desejado e o estado final é menor que $\theta/2$, o que significa que seu cosseno é maior que $\cos(\theta/2)$.

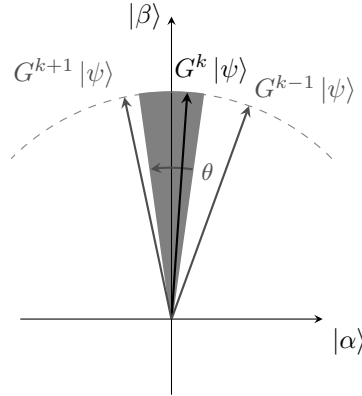


Figura 6.27: Projeção do estado final $G^k |\psi\rangle$ na direção do estado desejado $|\beta\rangle = |x_0\rangle$. Aplicar o operador G mais uma vez faz com que a projeção na direção desejada fique menor. O mesmo vale se o operador G for aplicado por menos de k vezes.

Com isso, tem-se a probabilidade de acerto estimada em

$$\begin{aligned}
 P_a &= \left\| |x_0\rangle \langle x_0| G^k |\psi\rangle \right\|^2 \\
 &= \left| \langle x_0 | G^k |\psi\rangle \right|^2 \\
 &= \left| \cos \theta (|x_0\rangle, G^k |\psi\rangle) \right|^2 \\
 &> |\cos \theta / 2|^2 \\
 &= \left(\frac{\sqrt{N-1}}{\sqrt{N}} \right)^2 \\
 &= \frac{N-1}{N}, \quad N = 2^n.
 \end{aligned}$$

Generalização

O algoritmo de Grover também funciona para o caso em que há mais de um elemento marcado, isto é, há x_0, x_1, \dots, x_{m-1} elementos tais que $f(x_0) = f(x_1) = \dots = f(x_{m-1})$ e os demais valores anulam f . É possível adaptar a análise do algoritmo para esse caso. A interpretação geométrica continua valendo, mas desta vez, tem-se

$$|\alpha\rangle = \frac{1}{\sqrt{N-m}} \sum_{\substack{x \in \mathbb{B}_n \\ f(x) = 0}} |x\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{m}} \sum_{\substack{x \in \mathbb{B}_n \\ f(x) = 1}} |x\rangle .$$

6.6.3 Algoritmo Clássico

Algoritmo Clássico Determinístico

Classicamente, se o teste é dado como uma caixa preta, em que não se sabe a estrutura interna de f , a maneira de se encontrar x_0 é por busca exaustiva. Para garantir que o item x_0 seja encontrado, deve-se, no pior caso, olhar todas as $N = 2^n$ entradas possíveis. Portanto são necessários N aplicações do teste f .

Algoritmo Clássico Probabilístico

Considere um algoritmo que sorteie aleatoriamente as entradas a serem testadas (e sem repetir entradas). Após o teste de k entradas, a probabilidade de que se tenha localizado o item desejado x_0 é de

$$P = \frac{k}{N} .$$

Para que haja probabilidade de encontrar a resposta seja maior que $1/2$, deve-se testar pelo menos por um número de vezes

$$k > \frac{N}{2} .$$

Desse modo, ainda deve-se aplicar o teste f da ordem de N vezes.

6.6.4 Comparação de Desempenho

O desempenho dos algoritmos são comparados na tabela abaixo.

Algoritmo	Desempenho (# aplicações de f)
Class. Det.	da ordem de $N = 2^n$ aplicações
Class. Prob.	da ordem de $N/2 = 2^n/2$ aplicações
Quântico	da ordem de $\sqrt{N} = 2^{n/2}$ aplicações

Tabela 6.6: Comparação de desempenho entre os algoritmos quântico, clássico determinístico e clássico probabilístico (com probabilidade de erro $< 50\%$) para o Problema de Grover. No algoritmo de Grover, o número de aplicações de f coincide com o número de aplicações da subrotina G .

Dessa forma, o algoritmo de Grover apresentaria ganho quadrático de desempenho em relação aos algoritmos clássicos, se a aplicação de f nos casos clássico e quântico forem equivalentes em termos de custos computacionais.

6.7 Algoritmo de Bernstein-Vazirani

Há duas versões do algoritmo de Bernstein-Vazirani, ambas formuladas para resolver o problema de mesmo nome. Este problema, como os outros problemas abordados, não tem perspectivas de aplicações, no entanto, ajuda a entender em que situações a computação quântica pode apresentar vantagens.

Neste texto, as duas versões do algoritmo são denominadas “versão XOR” e “versão fase”. Duas referências úteis para esses algoritmos são [21] (XOR) e [1] (fase). Ambas apresentam um ganho exponencial em relação à computação possível classicamente, com a versão fase mais eficiente que a versão XOR.

6.7.1 Problema de Bernstein-Vazirani

O objeto desta seção também é um problema de caixa preta, como nos outros algoritmos apresentados até então. Portanto é fornecida uma função booleana f , com determinada propriedade; neste caso, f é da forma $f(x) = a \cdot x$, com x e a vetores de bits, e o produto corresponde àquele análogo a um produto interno apresentado na observação 6.19 e reforçado a seguir. Neste problema, não se conhece a estrutura interna de f , ou seja, a é uma incógnita e só se conhece que a função f admite um a e que pode ser escrita como $f(x) = a \cdot x$. A demanda do problema é encontrar a com o menor número possível de aplicações de f .

Problema de Bernstein-Vazirani. *Seja $f: \{0, 1\}^n \rightarrow \{0, 1\}$ uma função booleana da forma*

$$f(x_1 \dots x_n) = a_1 \cdot x_1 \oplus \dots \oplus a_n \cdot x_n ,$$

que também poderia ser denotada⁶ por

$$f(x) = a \cdot x .$$

A função f é dada por uma caixa preta, sua estrutura interna não é conhecida. O problema pede para encontrar o vetor de bits $a = a_1 \dots a_n$.

Os algoritmos quânticos, bem como o procedimento clássico, são apresentados a seguir.

⁶Fazendo uma analogia com o produto interno em \mathbb{C}^n

6.7.2 Algoritmo de Bernstein-Vazirani (versão XOR)

O algoritmo quântico apresentado aqui conta com f representado como um oráculo XOR. Uma referência para o algoritmo a ser apresentado é [21]. Apesar de menos eficiente que a versão fase, esse algoritmo é abordado para ilustrar a técnica de *pós-seleção*, presente em outros algoritmos quânticos ainda não apresentados, como o algoritmo (quântico) HHL para resolução de sistemas lineares [9].

Algoritmo de Bernstein-Vazirani (versão XOR)

Entrada: $O_{\text{XOR}}(f)$ (oráculo XOR associado à função booleana f)

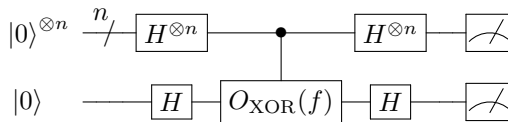
Procedimento:

- | | | |
|----------|---|--|
| etapa 0: | $ 0\rangle^{\otimes n} 0\rangle$ | preparação do estado inicial |
| etapa 1: | $\sum_x x\rangle 0\rangle$ | superposição de estados com $H^{\otimes n}$ no reg.1 |
| etapa 2: | $\sum_x x\rangle f(x)\rangle$ | aplicação de f (oráculo XOR) |
| etapa 3: | $\sum_x x\rangle H a \cdot x\rangle$ | aplicação de H no reg.2 |
| etapa 4: | $ 0\rangle 0\rangle + a\rangle 1\rangle$ | aplicação de $H^{\otimes n}$ no reg.1 |

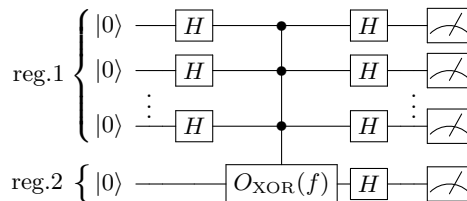
Saída: Mede-se o último qubit (reg.2). Caso o resultado seja 1, o reg.1 carregará o valor de a . Se o resultado da medição no reg.2 for 0, o algoritmo deve ser repetido até que se obtenha 1 nesse registrador. Isto significa fazer uma pós-seleção no reg.2

Circuito

Notação compacta:



Notação expandida:



Análise detalhada do algoritmo

Nesta explanação, utiliza-se diversas vezes a proposição 6.18, que fornece a expressão

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle ,$$

com $|x\rangle$ vetor da base computacional. Utiliza-se também a notação $|\text{reg.1}\rangle |\text{reg.2}\rangle$ para distinguir os dois conjuntos de qubits.

A primeira etapa do algoritmo resulta no estado

$$\begin{aligned} |\psi_1\rangle &= (H^{\otimes n} |0\rangle) |0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle |0\rangle . \end{aligned}$$

Obtém-se, ao aplicar f ,

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} O_{\text{XOR}}(f) |x\rangle |0\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle |f(x)\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle |a \cdot x\rangle . \end{aligned}$$

Na etapa 3, aplica-se a porta Hadamard ao registrador 2:

$$\begin{aligned} |\psi_3\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle (H |a \cdot x\rangle) \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle \left(\frac{|0\rangle + (-1)^{a \cdot x} |1\rangle}{\sqrt{2}} \right) \\ &= \frac{1}{\sqrt{2^n} \sqrt{2}} \sum_{x \in \mathbb{B}^n} |x\rangle |0\rangle + \frac{1}{\sqrt{2^n} \sqrt{2}} \sum_{x \in \mathbb{B}^n} (-1)^{a \cdot x} |x\rangle |1\rangle \\ &= \frac{1}{\sqrt{2}} (H^{\otimes n} |0\rangle) |0\rangle + \frac{1}{\sqrt{2}} (H^{\otimes n} |a\rangle) |1\rangle . \end{aligned}$$

Assim, ao aplicar-se a as portas Hadamard ao registrador 1, na etapa 4,

obtém-se que

$$\begin{aligned}
 |\psi_4\rangle &= \frac{1}{\sqrt{2}}(H^{\otimes n} |0\rangle) |0\rangle + \frac{1}{\sqrt{2}}(H^{\otimes n} |a\rangle) |1\rangle \\
 &= \frac{1}{\sqrt{2}}(H^{\otimes n} H^{\otimes n} |0\rangle) |0\rangle + \frac{1}{\sqrt{2}}(H^{\otimes n} H^{\otimes n} |a\rangle) |1\rangle \\
 &= \frac{1}{\sqrt{2}} |0\rangle |0\rangle + \frac{1}{\sqrt{2}} |a\rangle |1\rangle .
 \end{aligned}$$

Dessa forma, ao se fazer uma medida nos registradores 1 e 2, obtém-se, com probabilidade $\left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}$ o resultado $|a\rangle |1\rangle$. Faz-se uma *pós-seleção* no segundo registrador, e quando este resultado for 1, o outro registrador carregará consigo a resposta a do problema.

6.7.3 Algoritmo de Bernstein-Vazirani (versão fase)

O algoritmo de Bernstein-Vazirani, em sua versão fase, possui desempenho ligeiramente melhor que o da versão XOR, e não necessita de pós-seleção. O algoritmo a ser apresentado consta na referência [1], e resolve o problema objeto desta seção de maneira exata, sem probabilidade de falha, com apenas uma aplicação do oráculo de fase representando f .

Algoritmo de Bernstein-Vazirani (versão fase)

Entrada: $O_F(f)$ (oráculo de fase associado à função booleana f)

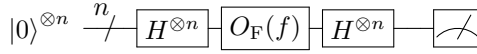
Procedimento:

etapa 0:	$ 0\rangle^{\otimes n} = 0\rangle$	preparação do estado inicial
etapa 1:	$H^{\otimes n} 0\rangle = \sum_x x\rangle$	superposição de estados com $H^{\otimes n}$
etapa 2:	$\sum_x (-1)^{a \cdot x} x\rangle = H^{\otimes n} a\rangle$	aplicação de f (oráculo de fase)
etapa 3:	$H^{\otimes n} H^{\otimes n} a\rangle = a\rangle$	aplicação de H

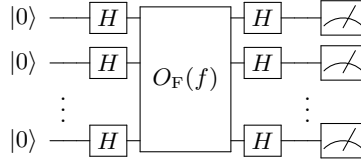
Saída: Mede-se o último qubit (reg.2). Caso o resultado seja 1, o reg.1 carregará o valor de a . Se o resultado da medição no reg.2 for 0, o algoritmo deve ser repetido até que se obtenha 1 nesse registrador.

Circuito

Notação compacta:



Notação expandida:



Análise detalhada do algoritmo

Nesta explanação, utiliza-se diversas vezes a proposição 6.18, que fornece a expressão

$$H^{\otimes n} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{B}_n} (-1)^{x \cdot y} |y\rangle ,$$

com $|x\rangle$ vetor da base computacional.

A primeira etapa do algoritmo resulta em

$$|\psi_1\rangle = H^{\otimes n} |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} |x\rangle .$$

Obtém-se, ao aplicar o oráculo de fase, que

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} O_F(f) |x\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} (-1)^{f(x)} |x\rangle \\ &= \frac{1}{\sqrt{2^n}} \sum_{x \in \mathbb{B}^n} (-1)^{a \cdot x} |x\rangle \\ &= H^{\otimes n} |a\rangle . \end{aligned}$$

Finalmente, aplicam-se as portas Hadamard, o que retorna

$$|\psi_3\rangle = H^{\otimes n} H^{\otimes n} |a\rangle = |a\rangle$$

Assim, uma medição no registrador fornecerá a incógnita a do problema.

6.7.4 Algoritmo Clássico

Algoritmo Clássico Determinístico

Na computação clássica, é possível resolver o problema de Bernstein-Vazirani aplicando-se a função booleana f aos n vetores de bits⁷

$$x = x_1x_2 \dots x_n = 100 \dots 0, \quad 010 \dots 0, \quad \dots, \quad 0 \dots 01.$$

De fato, se $f(x) = a \cdot x$, com a vetor de bits fixo, mas desconhecido, pode-se descobrir $a = a_1a_2 \dots a_n$ fazendo⁸

$$\begin{aligned} f(100 \dots 0) &= a_1 \cdot 1 \oplus a_2 \cdot 0 \oplus \dots \oplus a_n \cdot 0 = a_1 \oplus 0 \oplus \dots \oplus 0 = a_1 \\ f(010 \dots 0) &= a_1 \cdot 0 \oplus a_2 \cdot 1 \oplus \dots \oplus a_n \cdot 0 = 0 \oplus a_2 \oplus \dots \oplus 0 = a_2 \\ &\vdots \\ f(0 \dots 001) &= a_1 \cdot 0 \oplus \dots \oplus a_{n-1} \cdot 0 \oplus a_n \cdot 1 = 0 \oplus \dots \oplus 0 \oplus a_n = a_n \end{aligned}$$

Com isso, são necessárias n aplicações de f para se descobrir o vetor de bits a .

Algoritmo Clássico Probabilístico

Parece não ser possível melhorar o algoritmo clássico prescrito acima, pois cada aplicação de f nos fornece uma equação envolvendo a_1, \dots, a_n . Para encontrar todos os a_i s são necessárias n equações.

6.7.5 Comparação de Desempenho

Como nas outras seções, apresenta-se uma comparação de desempenho na tabela abaixo.

Algoritmo	Desempenho (# aplicações de f)
Class. Det.	n aplicações
Quântico (XOR)	da ordem de 2 aplicações
Quântico (fase)	1 aplicação

Tabela 6.7: Comparação de desempenho entre os algoritmos quântico, clássico determinístico e clássico probabilístico (com probabilidade de erro $< 50\%$) para o Problema de Bernstein-Vazirani.

⁷Os vetores de bits referidos são os com peso de Hamming 1. O peso de Hamming de um vetor de bits é o número de bits com valor igual a 1. Isso equivale à soma dos bits, considerando-os como números inteiros. O peso de Hamming de um vetor de n bits está entre 0 e n .

⁸Para lembrar, as operações \cdot e \oplus se referem à AND e XOR. O “produto interno” $a \cdot x$ refere-se a $(a_1 \text{ AND } x_1) \text{ XOR } \dots \text{ XOR } (a_n \text{ AND } x_n) = a_1 \cdot x_1 \oplus \dots \oplus a_n \cdot x_n$. Ver observação 6.19.

6.8 Transformada de Fourier Quântica - QFT

6.8.1 Transformada Discreta de Fourier - DFT

EXPLICAR o que é e fazer comparação?

6.8.2 Definição e Exemplos

6.8.3 Propriedades da QFT

6.8.3.1 A QFT é unitária

(fazer como proposição)

6.8.4 Circuito para QFT

6.8.4.1 QFT para $n = 1$ qubit

6.8.4.2 QFT para $n = 2$ qubits

6.8.4.3 QFT para n qubits

6.8.5 Algumas aplicações

6.9 Algoritmo de Shor

Conclusão

Considerações Finais

O presente trabalho trouxe uma abordagem da Computação Quântica em nível introdutório. Procurou-se abordar os principais pré-requisitos para tornar o texto autocontido. Procurou-se formular um material multidisciplinar, com enfoque tanto nos aspectos teóricos como nos desdobramentos atuais e nas expectativas de mercado. O objetivo do texto é trazer um material de fácil leitura para o iniciante nesse assunto, vindo de diversas áreas de Ciências Exatas.

Perspectivas

Pretende-se dar continuidade ao material, acrescentando-se novos tópicos, de forma a torná-lo uma referência útil aos iniciantes em Computação Quântica. Em particular, pretende-se acrescentar uma introdução à Transformada de Fourier Quântica e o Algoritmo de Shor, que é usado em um procedimento para encontrar fatores primos de um número inteiro com um desempenho superior aos algoritmos clássicos conhecidos. Tem-se também como perspectiva buscar algoritmos de interesse em problemas reais de Engenharia e áreas aplicadas.

Espera-se que em um futuro próximo, os estudantes de graduação da UFSC possam ter como opção em sua grade curricular um curso introdutório de Computação Quântica e que esse material possa contribuir para alcançar esse objetivo.

Bibliografia

- [1] Scott Aaronson. **Lecture Notes for Quantum Information Science**, <https://www.scottaaronson.com/qclec>. 2018.
- [2] Dorit Aharonov. **A simple proof that Toffoli and Hadamard are quantum universal**. *arXiv preprint quant-ph/0301040*, 2003.
- [3] Giuliano Benenti, Giulio Casati, and Giuliano Strini. **Principles of Quantum Computation and Information – Volume I: Basic Concepts**. World Scientific, 2004.
- [4] Matthew Brisse and Mark Horvath. **Quantum Computing: The Misunderstood and Feared Disruption**, Gartner, <https://www.gartner.com/webinar/3868072>. Acesso em abril de 2018, Apr 18 2018.
- [5] Steven Givant and Paul Halmos. **Introduction to Boolean Algebras**. Springer, 2009.
- [6] David J. Griffiths. **Introduction to Quantum Mechanics**. Prentice Hall, 1995.
- [7] Vanessa Pitirini Guarienti. **Computação Quântica Adiabática em Sistemas Relativísticos**. UFSC, 2016.
- [8] David Money Harris and Sarah L. Harris. **Digital Design and Computer Architecture**. Elsevier, 2nd edition, 2013.
- [9] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009.

- [10] Elon Lages Lima. **Álgebra Linear**. IMPA, 7 edition, 2006.
- [11] M. Morris Mano, Charles R. Kime, and Tom Martin. **Logic and Computer Design Fundamentals**. Pearson, 5th edition, 2015.
- [12] Jonas Maziero. **Álgebra Linear (Slides)**, UFSM, <https://sites.google.com/site/jonasmaziero/home/edu/topicos-em-ciencia-da-informacao-quantica>. Acesso em maio de 2018.
- [13] Todd K. Moon. **Error Correction Coding – Mathematical Methods and Algorithms**. Wiley, 2005.
- [14] Computer History Museum. **Exhibition – Birth of the Computer – ENIAC**, <http://www.computerhistory.org/revolution/birth-of-the-computer/4/78>. Acesso em maio de 2018.
- [15] Michael A. Nielsen and Isaac L. Chuang. **Quantum Computation and Quantum Information – 10th Anniversary Edition**. Cambridge University Press, 10th anv edition, 2010.
- [16] David A. Patterson and John L. Hennessy. **Computer Organization and Design: the hardware/software interface**. Elsevier, 5th edition, 2014.
- [17] Renato Portugal, Carlile Campos Lavor, Luiz Mariano Carvalho, and Nelson Maculan. **Uma Introdução à Computação Quântica**. SBMAC, 2 edition, 2012.
- [18] John Preskill. **Lecture Notes for Physics 229: Quantum Information and Computation**. 1998.
- [19] Peter Shor and Isaac Chuang. **Quantum Information Science I, part II**, edX, MIT, <https://courses.edx.org/courses/course-v1:MITx+8.370.2x+1T2018/course/>. acesso em abril de 2018.
- [20] Alfredo Steinbruch and Paulo Winterle. **Álgebra Linear**. Pearson, 2 edition, 1987.
- [21] IBM QX team. **Learning Parity with Noise**, IBM Quantum Experience, https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/004-Quantum_Algorithms/081-Learning_Parity_with_Noise.html. Acesso em maio de 2018.
- [22] Frank Vahid. **Digital Design with RTL Design, VHDL, and Verilog**. Wiley, 2nd edition, 2011.

-
- [23] Neal S. Widmer, Gregory L. Moss, and Ronald J. Tocci. **Digital Systems – Principles and Applications**. Pearson, 12th edition, 2017.

Apêndice A

Elementos de Computação Clássica

A.1 Introdução

Um computador digital é um sistema que pode seguir uma sequência de instruções, chamada programa, e que opera em um conjunto de informações. Os computadores digitais modernos são compostos de milhões a bilhões de transistores, que se agrupam em circuitos digitais. Para lidar com a complexidade desses sistemas, os circuitos são subdivididos em circuitos menores, que realizam funções específicas. Esses circuitos são considerados “caixas pretas”, em que se ignoram os detalhes internos, e são agrupados de forma a realizar funções mais sofisticadas.

A engenharia trabalha com *níveis de abstração*; cada nível corresponde a omitir detalhes internos dos subsistemas constituintes, ou da camada de abstração anterior. Uma discussão mais detalhada sobre as camadas de abstração do computador será realizada na seção seguinte.

Para que o computador consiga operar em um conjunto de informações, é necessário que essa informação seja traduzida, ou, codificada, de forma conveniente. O projeto dos computadores digitais se baseia em que as informações de entrada do sistema, e mesmo as instruções a serem seguidas, são codificadas em *bits*.

Os bits são variáveis que podem assumir apenas dois valores, rotulados de 0/1 ou Verdadeiro/Falso, por exemplo. No computador digital, a tensão elétrica é utilizada como bit; as tensões próximas a 0V são consideradas como bit 0 e as tensões próximas à tensão de alimentação do circuito

(normalmente 5V ou 3,3V), como bit 1.

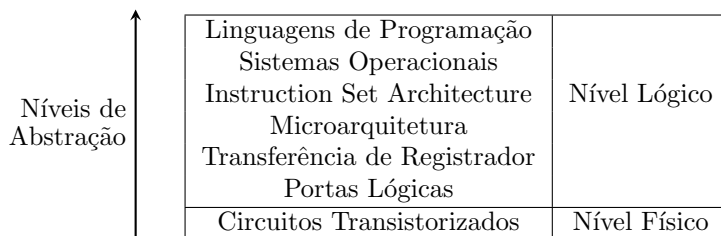
Nas seções seguintes alguns desses tópicos serão detalhados. A ênfase será nas ideias vinculadas aos Sistemas Digitais, no manejo da complexidade por meio das camadas de abstração e nos detalhes das camadas mais próximas da camada física, com o objetivo de passar a ideia de como um computador digital clássico funciona. A finalidade é, também, comparar esse paradigma de computação com as ideias que estão surgindo na área da Computação Quântica. As principais referências dessa seção são os livros [11], [23] e [22] de Sistemas Digitais e o livro [16] de Organização de Computadores.

A.2 Níveis de Abstração

Na engenharia, uma maneira de lidar com a complexidade de sistemas muito grandes é subdividi-los em subsistemas que possam ser descritos de maneira mais simples, omitindo detalhes internos. Componentes mais básicos são usados para projetar blocos que realizam funções simples. Esses blocos passam a ser descritos apenas pela sua função (como as saídas se comportam em relação às entradas), e passa-se a ignorar sua estrutura interna. Sistemas mais complexos podem ser projetados por meio desses blocos. A cada vez que se agrupa os sistemas em blocos e passa-se a ignorar sua estrutura interna, sobe-se um nível nas *camadas de abstração*. Quando se “abre” um sistema para analisar sua estrutura interna, passa-se à camada de abstração inferior.

Essa divisão em camadas de abstração permite que os diversos blocos do sistema sejam projetados de forma paralela. Além disso, o projeto de um bloco pode ser reaproveitado em outros momentos, no mesmo projeto ou em outros. Outra vantagem é que o sistema passa a ser visto como composto de uma quantidade relativamente pequena de subsistemas, em vez de ser visto como milhões de transistores, cujo funcionamento em conjunto seria virtualmente impossível de descrever diretamente.

A figura a seguir ilustra as camadas de abstração presentes no computador digital. Dependendo do autor, as camadas de abstração são nomeadas de maneira ligeiramente diferente ou são consideradas algumas subcamadas extra. Neste trabalho, a nomenclatura e as camadas de abstração consideradas seguirão a referência [11].



The diagram illustrates the levels of abstraction in computer systems. On the left, a vertical arrow points upwards, labeled "Níveis de Abstração". To the right of the arrow is a table with two columns. The first column lists various system components, and the second column groups them into two abstraction levels: "Nível Lógico" and "Nível Físico".

Linguagens de Programação Sistemas Operacionais Instruction Set Architecture Microarquitetura Transferência de Registrador Portas Lógicas	Nível Lógico
Circuitos Transistorizados	Nível Físico

Tabela A.1: Níveis de abstração – um método útil para lidar com a complexidade de sistemas. As camadas de baixo são mais próximas do nível físico e as de cima são mais abstratas. Fonte: [11] (adaptado).

A.3 Nível Lógico

O nível lógico refere-se à camada de abstração imediatamente acima da dos transistores. Os transistores são reunidos em *portas lógicas*. Nessa camada de abstração, os sinais de tensão na entrada e na saída são interpretados como bits, e as portas lógicas que operam esses bits simulam as funções lógicas como OR, AND, NOT, entre outras.

Nesse agrupamento em blocos os detalhes internos do circuito são ignorados.

A.3.1 Álgebra Booleana

As *variáveis booleanas* são variáveis que podem assumir apenas dois valores, rotulados como 0/1 ou Falso/Verdadeiro. Os bits são sinônimos de variáveis booleanas. As funções $f: \{0, 1\}^n \rightarrow \{0, 1\}^m$, que levam um conjunto de n bits em um conjunto de m bits, são chamadas *funções booleanas*. As funções booleanas podem ser especificadas por expressões matemáticas ou por uma tabela – a *tabela verdade* – listando todos os possíveis valores de entrada e a saída atribuída a cada valor de entrada.

Algumas funções booleanas elementares são chamadas de portas lógicas, ilustradas no tópico A.3.2 subsequente. As três operações básicas da Álgebra Booleana são $+$: $\{0, 1\}^2 \rightarrow \{0, 1\}$, \cdot : $\{0, 1\}^2 \rightarrow \{0, 1\}$ e \neg : $\{0, 1\} \rightarrow \{0, 1\}$, também chamadas de operações OR, AND e NOT, respectivamente.

A Álgebra Booleana pode ser interpretada como descrição de um sistema lógico em que há apenas dois valores lógicos – Falso/Verdadeiro ou 0/1 – e às proposições lógicas pode ser atribuído um e apenas um desses valores devido ao princípio lógico elementar do terceiro excluído.

Neste trabalho, o enfoque será mais voltado às aplicações em Sistemas Digitais. Um enfoque mais formal da álgebra booleana pode ser encon-

trado em [5], capítulo 2, em que se define uma álgebra booleana de forma axiomática.

A.3.2 Portas Lógicas

As portas lógicas são funções booleanas simples, blocos fundamentais dos circuitos digitais. As portas lógicas mais importantes são descritas resumidamente nas figuras a seguir.

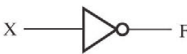
Nome	Símbolo	Equação	Tabela Verdade						
NOT		$F = \overline{X}$	<table><tr><th>X</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	X	F	0	1	1	0
X	F								
0	1								
1	0								

Figura A.1: Porta NOT.


Nome	Símbolo	Equação	Tabela Verdade															
AND		$F = X \cdot Y$	<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

Figura A.2: Porta AND.


Nome	Símbolo	Equação	Tabela Verdade															
OR		$F = X + Y$	<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	1
X	Y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

Figura A.3: Porta OR.


Nome	Símbolo	Equação	Tabela Verdade															
NAND		$F = \overline{X \cdot Y}$	<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	X	Y	F	0	0	1	0	1	1	1	0	1	1	1	0
X	Y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																

Figura A.4: Porta NAND.


Nome	Símbolo	Equação	Tabela Verdade															
			<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	X	Y	F	0	0	1	0	1	0	1	0	0	1	1	0
X	Y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
NOR		$F = \overline{X + Y}$																

Figura A.5: Porta NOR.


Nome	Símbolo	Equação	Tabela Verdade															
XOR		$F = X \oplus Y$	<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	0
X	Y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Figura A.6: Porta XOR (exclusive-OR).


Nome	Símbolo	Equação	Tabela Verdade															
XNOR		$F = \overline{X \oplus Y}$	<table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	X	Y	F	0	0	1	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Figura A.7: Porta XNOR (exclusive-NOR).

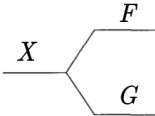
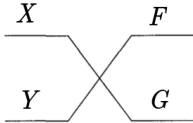
Nome	Símbolo	Tabela Verdade																				
FANOUT / COPY		<table><tr><th>X</th><th>F</th><th>G</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	X	F	G	0	0	0	1	1	1											
X	F	G																				
0	0	0																				
1	1	1																				
CROSSOVER / SWAP		<table><tr><th>X</th><th>Y</th><th>F</th><th>G</th></tr><tr><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	X	Y	F	G	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	1
X	Y	F	G																			
0	0	0	0																			
0	1	1	0																			
1	0	0	1																			
1	1	1	1																			

Figura A.8: Outras portas lógicas usadas implicitamente nos Sistemas Digitais: FANOUT / COPY e CROSSOVER / SWAP.

Qualquer sistema físico que se comporte de maneira a fornecer uma tabela verdade como as apresentadas acima pode ser considerado uma porta lógica.

A.3.3 Teoremas da Álgebra Booleana

Apresentam-se algumas identidades booleanas úteis para simplificação de expressões.

Teorema A.1 (Teoremas da Álgebra Booleana para uma variável).
Valem as seguintes identidades:

$X \cdot 0 = 0$	$X + 0 = X$
$X \cdot 1 = X$	$X + 1 = 1$
$X \cdot X = X$	$X + X = X$
$X \cdot \overline{X} = 0$	$X + \overline{X} = 1$

Demonstração. As igualdades se verificam testando todos os casos:

$$\begin{array}{ll}
 X \cdot 0 = 0 : \begin{cases} 0 \cdot 0 = 0 & (X = 0) \\ 1 \cdot 0 = 0 & (X = 1) \end{cases} & X + 0 = 0 : \begin{cases} 0 + 0 = 0 & (X = 0) \\ 1 + 0 = 1 & (X = 1) \end{cases} \\
 X \cdot 1 = X : \begin{cases} 0 \cdot 1 = 0 & (X = 0) \\ 1 \cdot 1 = 1 & (X = 1) \end{cases} & X + 1 = 1 : \begin{cases} 0 + 1 = 1 & (X = 0) \\ 1 + 1 = 1 & (X = 1) \end{cases} \\
 X \cdot X = X : \begin{cases} 0 \cdot 0 = 0 & (X = 0) \\ 1 \cdot 1 = 1 & (X = 1) \end{cases} & X + X = X : \begin{cases} 0 + 0 = 0 & (X = 0) \\ 1 + 1 = 1 & (X = 1) \end{cases} \\
 X \cdot \overline{X} = 0 : \begin{cases} 0 \cdot 1 = 0 & (X = 0) \\ 1 \cdot 0 = 0 & (X = 1) \end{cases} & X + \overline{X} = 1 : \begin{cases} 0 + 1 = 1 & (X = 0) \\ 1 + 0 = 1 & (X = 1) \end{cases}
 \end{array}$$

□

Teorema A.2 (Teoremas da Álgebra Booleana para várias variáveis).
Valem as seguintes identidades:

$$\begin{array}{ll}
 \text{Associatividade} & X + (Y + Z) = (X + Y) + Z \\
 & (XY)Z = X(YZ) \\
 \text{Comutatividade} & X + Y = Y + X \\
 & XY = YX \\
 \text{Distributividade} & X(Y + Z) = XY + XZ \\
 & (X + Y)Z = XZ + YZ \\
 & (X + Y)(Z + W) = XZ + XW + YZ + YW \\
 \text{Outras} & X + XY = X \\
 & X + \overline{X}Y = X + Y \\
 & \overline{X} + XY = \overline{X} + Y
 \end{array}$$

Demonstração. A verificação se dá atribuindo valores às variáveis ou escrevendo a tabela verdade dos dois lados da equação e verificando que o resultado é o mesmo. Pode-se usar o teorema A.1 para facilitar. Por exemplo, verifica-se a identidade do $X(Y + Z) = XY + XZ$:

Para $X = 0$: $0(Y + Z) = 0 = 0Y + 0Z$.

Para $X = 1$: $1(Y + Z) = Y + Z = 1Y + 1Z$.

□

Teorema A.3 (Teoremas DeMorgan).

Valem as seguintes identidades booleanas:

$$\begin{aligned}
 \overline{X + Y} &= \overline{X} \cdot \overline{Y} \\
 \overline{X \cdot Y} &= \overline{X} + \overline{Y}
 \end{aligned}$$

Demonstração.

Mostrando $\overline{X + Y} = \overline{X} \cdot \overline{Y}$:

X	Y	$X + Y$	$\overline{X + Y}$	\overline{X}	\overline{Y}	$\overline{X} \cdot \overline{Y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

Os valores das colunas $\overline{X + Y}$ e $\overline{X} \cdot \overline{Y}$ coincidem, portanto vale a igualdade.

Mostrando $\overline{X \cdot Y} = \overline{X} + \overline{Y}$:

X	Y	$X \cdot Y$	$\overline{X \cdot Y}$	\overline{X}	\overline{Y}	$\overline{X} + \overline{Y}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

Como os valores das colunas $\overline{X \cdot Y}$ e $\overline{X} + \overline{Y}$ coincidem, igualdade é válida. \square

A.3.4 Universalidade das Portas Lógicas Clássicas

Com apenas algumas das portas lógicas apresentadas em A.3.2 pode-se compor qualquer função booleana.

Teorema A.4 (Universalidade das portas OR, AND e NOT).

Uma função booleana $f: \{0, 1\}^m \rightarrow \{0, 1\}^n$ qualquer pode ser implementada por uma composição das portas lógicas OR, AND e NOT (além das portas SWAP e FANOUT).

Demonstração. Considere uma função booleana $f: \{0, 1\}^m \rightarrow \{0, 1\}^n$ qualquer. Basta fazer a demonstração para $n = 1$. Considerando que esse caso já esteja demonstrado, e visto que pode-se usar a porta FANOUT para copiar cada uma das m entradas o número de vezes que for necessário, pode-se implementar todas as n funções que retornam apenas 1 bit: $f_i: \{0, 1\}^m \rightarrow \{0, 1\}$, $i = 1, 2, \dots, n$. O caso analisado será, então, o de uma função que retorna apenas $n = 1$ bit na saída.

Considere a seguinte notação. O vetor de bits $(A_0, A_1, \dots, A_{m-1})$, que se pode representar pela justaposição $A_0 A_1 \dots A_{m-1}$, pode assumir os 2^m valores $0 \dots 00$, $0 \dots 01$, até $1 \dots 11$. Esses vetores podem ser identificados com a representação de números inteiros sem sinal na base 2 conforme ilustra a tabela abaixo.

$A_0 A_1 \dots A_{m-1}$	Número inteiro
0...00	$0 \cdot 2^{m-1} + \dots + 0 \cdot 2^1 + 0 \cdot 2^0 = 0$
0...01	$0 \cdot 2^{m-1} + \dots + 0 \cdot 2^1 + 1 \cdot 2^0 = 1$
0...10	$0 \cdot 2^{m-1} + \dots + 1 \cdot 2^1 + 0 \cdot 2^0 = 2$
0...11	$0 \cdot 2^{m-1} + \dots + 1 \cdot 2^1 + 1 \cdot 2^0 = 3$
\vdots	\vdots
1...11	$1 \cdot 2^{m-1} + \dots + 1 \cdot 2^1 + 1 \cdot 2^0 = 2^m - 1$

Tabela A.2: Correspondência entre vetor de bits $A_0 A_1 \dots A_{m-1}$ e o sub-conjunto de números inteiros sem sinal $\{0, 1, 2, \dots, 2^m - 1\}$.

Com essa correspondência, passa-se a identificar o vetor de bits com o número inteiro sem sinal associado. Dessa forma, pode-se denotar $f(0 \dots 11)$ por $f(3)$, por exemplo.

Seja $m_i: \{0, 1\}^m \rightarrow \{0, 1\}$, com $i = 0, 1, \dots, (2^n - 1)$, dada por $m_i(i) = 1$ e $m_i(j) = 0$ se $i \neq j$. Essas funções são chamadas *minitermos*, e assumem o valor 1 para exatamente um vetor de bits de entrada.

Seja $I = \{i: f(i) = 1\}$ o conjunto de entradas em que f assume o valor 1. Pode-se decompor f como a soma (OR) abaixo:

$$f = \sum_{i \in I} m_i.$$

Essa soma adquire valor 1 exatamente quando algum dos minitermos m_i assume 1. Como os minitermos considerados são os associados às entradas em que f assume o valor 1, a soma assume 1 exatamente nas mesmas entradas em que f assume o valor 1.

	A	B	C	f		A	B	C	m_2	m_4	m_7
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	1	0	0	0
2	0	1	0	1	2	0	1	0	1	0	0
3	0	1	1	0	3	0	1	1	0	0	0
4	1	0	0	1	4	1	0	0	0	1	0
5	1	0	1	0	5	1	0	1	0	0	0
6	1	1	0	0	6	1	1	0	0	0	0
7	1	1	1	1	7	1	1	1	0	0	1

$$\text{Portanto: } f = m_2 + m_4 + m_7$$

Figura A.9: Exemplo: $m = 3$ bits e decomposição de f .

Mas cada minitermo m_i pode, por sua vez, ser implementado com portas AND e NOT da seguinte forma. Considere i fixo e seja $i = A_0A_1 \dots A_{m-1}$, conforme a notação adotada. Alguns desses m bits assumem valor 0 e o restante, o valor 1. Denote por A_k os bits que assumem valor 0 e A_l os bits que assumem o valor 1, para certos conjuntos de índices K e L .

Agora seja $j = B_0B_1 \dots B_{m-1}$, que fará papel da entrada da função m_i . Considere o produto (AND) abaixo:

$$\prod_{k \in K} \overline{B_k} \cdot \prod_{l \in L} B_l$$

Essa função assume o valor 1 apenas quando todos os termos do produto (AND) valem 1. Isso ocorre apenas para $j = i$, isto é, para $A_0 = B_0, \dots, A_{m-1} = B_{m-1}$. Portanto:

$$\prod_{k \in K} \overline{B_k} \cdot \prod_{l \in L} B_l = m_i(B_0 \dots B_{m-1}).$$

	A	B	C	\overline{A}	B	\overline{C}	$\overline{A}B\overline{C}$
0	0	0	0	1	0	1	0
1	0	0	1	1	0	0	0
2	0	1	0	1	1	1	1
3	0	1	1	1	1	0	0
4	1	0	0	0	0	1	0
5	1	0	1	0	0	0	0
6	1	1	0	0	1	1	0
7	1	1	1	0	1	0	0

Para entrada 2: $\overline{A}B\overline{C} = 1 \cdot 1 \cdot 1 = 1$

Para as outras entradas: há pelo menos um 0 no produto (AND), o que faz com que o resultado fique 0.

Portanto: $m_2 = \overline{A}B\overline{C}$.

Figura A.10: Exemplo: $m = 3$ bits e obtenção do minitermo m_2 .

Dessa maneira, os minitermos m_i e a função f podem ser realizados com portas OR, AND e NOT.

□

Observação A.5. Pelas leis DeMorgan, é possível escrever a porta OR em termos das portas NOT e AND

$$x + y = \overline{\overline{x + y}} = \overline{\overline{x} \cdot \overline{y}}$$

e é possível escrever a porta AND em termos das portas NOT e OR fazendo

$$x \cdot y = \overline{\overline{x} + \overline{y}}.$$

Dessa forma, pode-se excluir a porta OR ou a AND no teorema A.4 e continua-se obtendo um conjunto de portas universal.

Teorema A.6 (Universalidade da porta NAND).

Uma função booleana $f: \{0,1\}^m \rightarrow \{0,1\}^n$ qualquer pode ser implementada por uma composição de portas lógicas NAND (além das portas SWAP e FANOUT).

Demonstração. Pelo teorema A.4, qualquer função booleana f pode ser implementada por portas lógicas OR, AND e NOT. Basta então mostrar que é possível obter as portas OR, AND e NOT a partir da porta NAND. Isso é possível, como se pode observar a seguir:

Porta NOT:

$$\text{NOT}(A) = \overline{A} = \overline{A \cdot A} = \text{NAND}(A, A)$$

Porta AND:

$$\text{AND}(A, B) = A \cdot B = \overline{\overline{A \cdot B}} = \text{NOT}(\text{NAND}(A, B))$$

Porta OR:

$$\text{OR}(A, B) = A + B = \overline{\overline{A + B}} = \overline{\overline{A} \cdot \overline{B}} = \text{NAND}(\text{NOT}(A), \text{NOT}(B))$$

Como a porta NOT pode ser construída usando portas NAND, as portas AND e NOT, que utilizam NAND e NOT também podem ser construídas apenas com portas NAND.

□

A.3.5 Somadores e Unidade Lógica/Aritmética

Com portas lógicas é possível implementar diversos componentes que realizam funções mais complexas. Alguns exemplos são somadores, subtrações, comparadores de igualdade, entre outros. Nessa seção, o somador de n bits é construído a partir de portas lógicas. Também é apresentada, a título de curiosidade, uma versão simplificada da Unidade Lógica/Aritmética, um dos principais componentes do bloco operativo de um processador.

Half Adder

O meio somador, também chamado *half adder*, é um componente que realiza a soma de duas entradas A e B , de 1 bit, e disponibiliza o resultado da soma na variável S , de 1 bit, e fornece um bit “vai um” (*carry out*), denotado por C_{out} . A tabela verdade desejada para esse sistema é dada na figura A.11, apresentada a seguir.

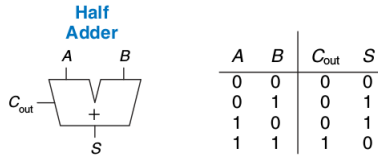


Figura A.11: Tabela verdade do half adder. Fonte: [8], p. 240

Pode-se perceber que as expressões booleanas para as saídas S e C_{out} são dadas pelas portas lógicas XOR e AND, respectivamente:

$$S = A \oplus B ,$$

$$C_{out} = A \cdot B .$$

Dessa forma, pode-se construir um circuito half adder de acordo com o esquemático da figura A.12.

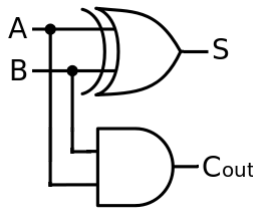


Figura A.12: Uma realização do half adder.

Full Adder

O somador completo de 1 bit, também chamado *full adder*, é um componente que realiza a soma de duas entradas de 1 bit, A e B . É interessante ter uma entrada *carry in* (C_{in}), a ser somada com A e B para modelar o “vai um” que entra na casa binária em questão. A saída é o resultado da soma, S e o “vai um” para a próxima casa binária é denotado por *carry*

out (C_{out}). A tabela verdade desejada para esse sistema é dada na figura A.13.

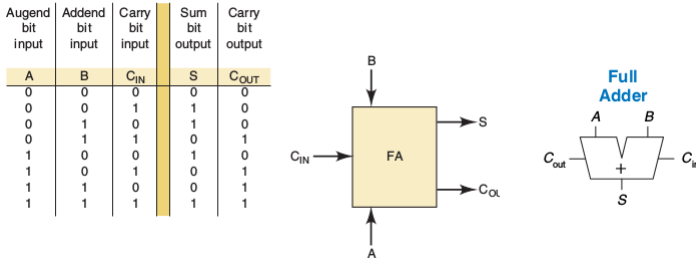


Figura A.13: Tabela verdade do full adder. Fonte: [23], p. 674

Para implementar o somador completo, podemos extrair as equações booleanas para as saídas S e C_{out} em função das entradas A , B e C_{in} . Escrevendo as entradas como soma de minitermos (como feito no Teorema A.4), obtemos:

$$\begin{aligned}
 S &= \overline{A} \overline{B} C_{in} + \overline{A} B \overline{C_{in}} + A \overline{B} \overline{C_{in}} + A B C_{in} \\
 &= \overline{A} (\overline{B} C_{in} + B \overline{C_{in}}) + A (\overline{B} \overline{C_{in}} + B C_{in}) \\
 &= \overline{A} (B \oplus C_{in}) + A (\overline{B \oplus C_{in}}) \\
 &= A \oplus B \oplus C_{in} ,
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= \overline{A} B C_{in} + A \overline{B} C_{in} + A B \overline{C_{in}} + A B C_{in} \\
 &= (\overline{A} B + A \overline{B}) C_{in} + A B (C_{in} + \overline{C_{in}}) \\
 &= (A \oplus B) C_{in} + A B .
 \end{aligned}$$

Portanto, pode-se implementar o full adder com o circuito da figura A.14. Aproveita-se uma porta XOR no cálculo das duas expressões.

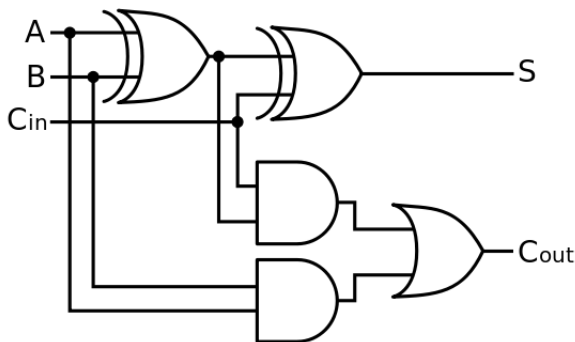


Figura A.14: Uma realização do full adder.

Somador de n bits

Com o full adder, é possível implementar um somador de n bits. Esse componente realiza a soma de n bits, interpretados como números inteiros sem sinal (da mesma forma que no teorema A.4) e disponibiliza o resultado da soma em n bits e um bit carry out, que indica se houve *overflow*, isto é, se a soma ultrapassou o valor máximo possível de ser representado pelos n bits (que seria $2^n - 1$ no caso de inteiros sem sinal). A figura abaixo mostra o símbolo para um somador de n bits e uma implementação usando full adders.

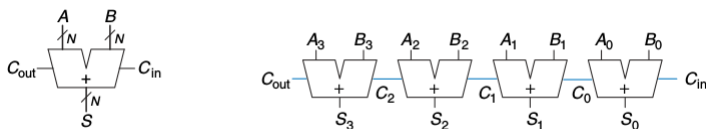


Figura A.15: Símbolo para somador de n bits. Implementação de um somador de 4 bits usando 4 full adders. Fonte: [8], p. 240

Com algumas modificações, o somador de n bits pode realizar subtrações também. Para tanto, as entradas devem ser interpretadas como inteiros com sinal. A representação de inteiros com sinal se dá por *complemento de 2*. Essa representação não será abordada neste trabalho, podendo ser encontrada em mais detalhes nas referências [8], p. 16-19, e [23], p. 343-355.