Notas em Matemática Aplicada e-ISSN 2236-5915

Volume 65, 2012

Sociedade Brasileira de Matemática Aplicada e Computacional

2012

A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC em 2003, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa a publicar, também, livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7), as figuras em eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja todos os títulos publicados nesta série na página
http://www.sbmac.org.br/notas.php

Sociedade Brasileira de Matemática Aplicada e Computacional

2012

CÓDIGOS QUÂNTICOS CORRETORES DE ERROS

Renato Portugal
portugal@lncc.br

Coordenação de Ciência da Computação
Laboratório Nacional de Computação Científica – LNCC
Ministério da Ciência, Tecnologia e Inovação – MCTI

Demerson Nunes Gonçalves
demerson.goncalves@ucp.br

Centro de Engenharia e Computação
R. Barão do Amazonas, 124, Centro, 25.685-070, Petrópolis, RJ
Universidade Católica de Petrópolis – UCP

Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil
2012

Coordenação Editorial: Elbert Einstein Nehrer Macau

Coordenação Editorial da Série: Rosana Sueli da Motta Jafelice

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Catalogação elaborada pela Biblioteca do IBILCE/UNESP
Bibliotecária: Maria Luiza Fernandes Jardim Froner

Portugal, Renato
Códigos Quânticos Corretores de Erros - São Carlos, SP : SBMAC, 2012, 86 p., 20.5 cm - (Notas em Matemática Aplicada; v. 65)

e-ISBN 978-85-8215-016-0

1. Códigos Estabilizadores 2. Códigos Não-Aditivos
3. Códigos CSS 4. Códigos CWS 5. Computação Quântica
6. Segurança da Informação Quântica
I. Portugal, Renato II. Gonçalves, Demerson N.
III. Título. IV. Série

CDD - 51

Dedicamos aos companheiros de jornada
na estrada da Computação Quântica.

Conteúdo

Prefácio

A Computação Quântica é uma área bem estabelecida, com uma grande quantidade de resultados teóricos dentro do contexto da Teoria da Computação, assim como resultados em Engenharia e Física, que permitem a construção de protótipos de *hardwares quânticos*. Atualmente, o computador quântico mais avançado tem na ordem de uma centena de bits quânticos (*qubits*).

A grande maioria das pessoas, que não é da área e ouve falar do computador quântico, espera que o desenvolvimento do *hardware* obedeça à famosa *Lei de Moore*, válida na construção do computador clássico por cerca de cinquenta anos. Muitas dessas pessoas se decepcionam ao saber da enorme dificuldade teórica e tecnológica para domar e controlar memórias do tamanho de alguns átomos, onde a leis quânticas dominam em sua plenitude. A construção do computador quântico requer que a barreira semi-clássica, que norteia a construção dos semicondutores usados nos computadores clássicos, seja rompida e algo equivalente totalmente quântico seja desenvolvido para que as operações lógicas elementares possam ser implementadas.

O computador clássico é extremamente estável no seu processamento. Dependendo do cálculo, uma única inversão de bit poderia invalidar todo o processo. Porém, sabemos que longos cálculos, que requerem bilhões de inversões de bits, são feitos sem problemas. Os computadores clássicos são insensíveis a erros porque seus componentes básicos são estáveis. Pense por exemplo em um computador mecânico. Seria muito raro um dispositivo mecânico inverter de posição por si só, principalmente se colocarmos uma mola para mantê-lo estável nas posições desejadas. O mesmo é válido para dispositivos

eletrônicos, que permanecem em seu estado até que um pulso elétrico de potência suficiente inverta sua posição. Os dispositivos eletrônicos são construídos para funcionarem em um nível de potência bem superior ao ruído e o ruído é mantido baixo por dissipação de calor para o ambiente. Isso mostra que técnicas de correção de erros não são importantes no processamento em dispositivos clássicos até o momento.

Os computadores quânticos usam a superposição quântica de estados para fazer um novo tipo de processamento. O entendimento e a interpretação dessa superposição é um desafio, pois como podemos entender uma partícula girando para no sentido horário e anti-horário ao mesmo tempo? Como regra geral, um dispositivo físico precisa ser isolado do ambiente para que a superposição não seja perdida. Isolar sistemas físicos do seu ambiente é uma tarefa ingrata. Partículas ultrarelativísticas e ondas gravitacionais passam por qualquer bloqueio, penetram em sistemas "isolados", pegam informações e as levam para fora do sistema. Esse processo é equivalente a uma medida de um observável quântico, que em geral provoca o colapso da superposição e freia o computador quântico, tornando-o equivalente ao clássico. As técnicas de amplificação de sinal e dissipação de ruídos não podem ser aplicadas em dispositivos quânticos da mesma forma que são usadas nos dispositivos clássicos. O processamento quântico é unitário e reversível. A unitaridade nesse contexto significa que as probabilidades envolvidas no processo têm medida unitária. A replicação de informação quântica é não-unitária e processos dissipativos são irreversíveis. Como, na prática, o isolamento nunca é total, é necessário lançar mão de técnicas de correção de erros.

O texto está organizado em três capítulos e um apêndice. No primeiro, é apresentada uma introdução sobre teoria de correção quântica de erros. No segundo capítulo, a ênfase é dada aos códigos aditivos, onde a teoria quântica de correção de erros é dada através do formalismo estabilizador. Os códigos corretores quânticos aditivos ocupam a maior parte do material e são os mais estudados na literatura, no entanto, eles não são os códigos ótimos no caso geral. Nos últimos anos, uma técnica de construção de códigos corretores não-aditivos foi desenvolvida e diversos códigos não-aditivos mais eficientes foram ob-

tidos. Reservamos a parte final deste curso para a teoria dos códigos CWS. A teoria dos códigos quânticos usa extensivamente a Teoria de Grupos Finitos, que é uma área da Álgebra Abstrata. Para tornar o curso auto-suficiente nesse aspecto, acrescentamos um apêndice que cobre a teoria de grupos básica. Diversos outros pré-requisitos são necessários. É importante que o leitor tenha um conhecimento básico de Computação Quântica que pode ser obtido na Ref. [22]. Tanto a Computação Quântica quanto a Teoria dos Códigos Quânticos usam a Álgebra Linear como linguagem base. É importante que o leitor esteja familiarizado com os resultados da Álgebra Linear descritos no apêndice da Ref. [21]. Uma descrição dos postulados da Mecânica Quântica também pode ser encontrada na Ref. [21]. Uma versão preliminar deste curso foi apresentada no 1° *Encontro em Teoria dos Códigos e Criptografia* na UFABC. A Ref. [20] também pode ser de grande valia para que o leitor adquira conhecimento suficiente para o entendimento deste curso, em especial, no que tange a circuitos quânticos.

Críticas e sugestões por parte dos leitores são bem-vindas e podem ajudar em futuras edições melhoradas de nosso trabalho. Finalmente, agradecemos o apoio da Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), o contínuo apoio do CNPq, principalmente através dos editais dos "Grandes Desafios da Computação" e o apoio da CAPES e FAPERJ.

Capítulo 1

Correção Quântica de Erros

A teoria dos *códigos quânticos* de correção de erros estende as noções básicas dos *códigos clássicos* de correção de erros. Uma diferença marcante reside no fato de que os erros quânticos estão muito mais presentes do que no caso clássico. Não é possível implementar um hardware quântico sem lidar com correções de erros. Os *estados quânticos* facilmente entram em *descoerência* devido a influências do *sistema macroscópico* sobre o sistema quântico. A correção de erros é possível quando armazenamos a informação quântica de forma redundante. Os métodos quânticos servem para (1) garantir o funcionamento correto dos algoritmos quânticos, (2) enviar mensagens codificadas por um canal quântico de forma resistentes a erros e (3) armazenar dados quânticos de forma segura.

1.1 Modelo Padrão

Vamos supor que Alice queira enviar uma mensagem para Beto através de um canal, que não é perfeito. No caso clássico, Alice teria uma mensagem composta de uma *string* de caracteres binários. O modelo mais simples de canal clássico, que é chamado de simétrico, admite uma probabilidade de erro p do bit 0 ser convertido para o bit 1 e vice-versa. Alice codifica a mensagem usando redundância e envia pelo canal, um bit de cada vez. Beto recebe uma *string* que pode ter sido

adulterada, ele faz a *análise de síndrome* para determinar se houve erro e tenta recuperar a mensagem original. Se a probabilidade p for suficientemente pequena ou se a redundância usada na codificação for suficientemente grande, Beto pode ter sucesso em recuperar a mensagem completamente. Os bons códigos maximizam a chance de sucesso sem usar um excesso de redundância.

A *codificação* mais simples replica os bits originais da seguinte forma: $0 \to 0_L$ onde $0_L = 000$ e $1 \to 1_L$ onde $1 = 111$. A mensagem triplica de tamanho. Por exemplo, se Alice quer enviar a informação 101, ela envia de fato 111000111. Beto recebe a mensagem codificada, verifica quais blocos de 3 bits não tem o formato 000 ou 111 e usa o método de "voto de maioria" para corrigir os bits "errados". Por exemplo, Beto pode ter recebido 101000111 e ele percebe que o erro ocorreu no primeiro bloco e ele inverte o segundo bit para obter a mensagem codificada correta. Após a codificação, a probabilidade de ocorrer um erro irrecuperável nos bits lógicos 0_L ou 1_L é quadraticamente menor, pois é necessário a inversão de pelo menos 2 bits para que o método do voto de maioria não funcione de forma correta.

No caso quântico, Alice vai usar *bits quânticos* (qubits) para escrever a mensagem. Ela pode usar tanto os estados $|0\rangle$ e $|1\rangle$ como *estados em superposição* $|\psi\rangle = a|0\rangle + b|1\rangle$. A mensagem é enviada pelo canal quântico, um qubit de cada vez. Um possível erro que pode acontecer no canal é *inversão de qubit*: $|0\rangle \to |1\rangle$ e vice-versa. Mas esse não é o único tipo de erro. A *inversão de fase*, $|0\rangle \to |0\rangle$ e $|1\rangle \to -|1\rangle$, é tão problemática quanto a inversão de qubit. Por exemplo, se o qubit da mensagem for $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, após a inversão de fase, o qubit é transformado no estado $|\psi'\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$, que é ortogonal ao primeiro.

No caso geral, o erro é um operador linear genérico de 1 qubit. Note que há um continuum de possibilidades. Alice deve codificar a mensagem e enviar pelo canal quântico. Beto deve medir cada qubit que for recebendo, ou esperar e formar blocos antes de medir. Essas medições devem ter o intuito de detectar os erros para corrigí-los e reproduzir a mensagem original. A mensagem original não deve ser destruída no processo de medição. Isso tem que ser feito criteriosa-

mente, pois sabemos que, no caso geral, *medidas* perturbam o estado quântico. Temos que obter informações apenas sobre os erros e não sobre a mensagem original. Após ter em mãos a mensagem original, Beto pode dar o destino que quiser à mensagem.

1.2 Código de Repetição de 3 Qubits

O exemplo mais simples de um código quântico é o *código de repetição* de 3 qubits para corrigir inversão de 1 qubit. Esse código também é conhecido como *código de inversão de qubit*. Suponha que um qubit da mensagem seja $|\psi\rangle = a|0\rangle + b|1\rangle$. A codificação visa substituir os qubits $|0\rangle$ e $|1\rangle$ por

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle = |000\rangle \\ |1\rangle &\rightarrow |1_L\rangle = |111\rangle. \end{aligned}$$

Esse código corrige no máximo um erro de inversão de qubit nos estados lógicos $|0_L\rangle$ e $|1_L\rangle$.

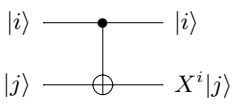

Figura 1.1: Circuito da porta CNOT. As variáveis i e j assumem os valores 0 ou 1. X é a matriz de Pauli que inverte um qubit.

A *porta CNOT* é a porta adequada para replicar os qubits $|0\rangle$ e $|1\rangle$. O funcionamento do CNOT está descrito na Fig. 1.1. O primeiro qubit (inicialmente no estado $|i\rangle$) é o controle e o segundo qubit (inicialmente no estado $|j\rangle$) é o alvo. Se o valor de i for 0, a saída será igual a entrada. Se o valor de i for 1, o controle é ativado e a saída do qubit alvo será modificada pela *matriz de Pauli* X. O funcionamento da porta foi descrito para a *base computacional*. A linearidade deve ser usada para se obter a saída quando a entrada são estados em superposição. A

porta CNOT é um operador unitário cuja representação matricial é

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1.2.1}$$

Pela Fig. 1.1, vemos que se $j = 0$, a saída será $|i\rangle|i\rangle$. Portanto, a porta CNOT faz uma cópia do estado do primeiro registrador, quando ele é um estado da base computacional. Se colocarmos o estado $|\psi\rangle = a|0\rangle + b|1\rangle$ como entrada para o qubit de controle e o estado $|0\rangle$ como entrada para o qubit alvo, obtemos a saída $a|00\rangle + b|11\rangle$ por linearidade.

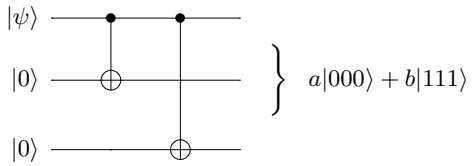

Figura 1.2: Circuito para a codificação de 3 qubits, onde $|\psi\rangle = a|0\rangle + b|1\rangle$.

Para fazer a codificação de 3 qubits, usamos duas portas CNOTs como no circuito da Fig. 1.2. Esse circuito faz a substituição dos estados $|0\rangle$ e $|1\rangle$ do estado original para os estados $|0_L\rangle$ e $|1_L\rangle$. A etapa final da *decodificação* é feita com o circuito transposto conjugado, que nesse caso coincide com o circuito original.

Após o estado $|\psi\rangle$ ser codificado, Alice envia os 3 qubits pelo canal quântico para Beto. Esses qubits podem ser enviados um por vez. Note que $|\psi\rangle$ está *emaranhado* no caso geral. Não há problemas em enviar os qubits separadamente, pois o *emaranhamento* é preservado mesmo quando os qubits estão afastados uns dos outros. Em geral, os canais têm todo tipo de imperfeição, porém vamos supor que ocorre apenas inversão de qubit no máximo em um dos qubits.

Beto recebe os qubits e faz a análise de síndrome fazendo medidas em cascata dos *observáveis* Z_1Z_2 e Z_2Z_3. A decomposição de Z_1Z_2

em *projetores* é

$$Z_1 Z_2 = P_+ - P_-, \tag{1.2.2}$$

onde

$$P_+ = (|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I_2 \tag{1.2.3}$$
$$P_- = (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I_2. \tag{1.2.4}$$

As probabilidades associadas aos valores ± 1 são $p_\pm = \langle\psi|P_\pm|\psi\rangle$, respectivamente. A medida do observável $Z_1 Z_2$ retorna $+1$ se os valores dos 2 primeiros qubits forem iguais e -1 se forem diferentes, análogo para $Z_2 Z_3$ com relação aos 2 últimos qubits. Se não houve nenhuma inversão de qubits, ambas medidas retornam $+1$. Se houve inversão do primeiro qubit, a primeira medida retorna -1 e a segunda $+1$. Se houve inversão do segundo qubit, ambas medidas retornam -1. Finalmente, se houve inversão do terceito qubit, a primeira medida retorna $+1$ e a segunda -1. A Tabela 1.1 resume todo o processo.

Erro	Síndrome	Correção
I	1, 1	I
X_1	-1, 1	X_1
X_2	-1, -1	X_2
X_3	1, -1	X_3

Tabela 1.1: Erros de inversão de 1 qubit no código de 3 qubits, análise de síndrome e os operadores de correção. Os resultados das medidas correspondem a medidas em cascata dos observáveis $Z_1 Z_2$ e $Z_2 Z_3$.

A correção é feita aplicando-se a matriz de Pauli X no qubit invertido. A etapa final da decodificação é feita com o mesmo circuito da codificação. Beto deve descartar os dois qubits auxiliares. O qubit restante estará no estado $|\psi\rangle$, como preparado por Alice.

O processo de codificação e decodificação (síndrome, recuperação e aplicação do circuito inverso da codificação) foi descrito com sucesso no código de 3 qubits. Todo o processo pode ser colocado na forma de um circuito, como mostrado na Fig. 1.3. Note que as medições estão representadas por caixas arredondadas com o observável especificado no seu interior. Os resultados clássicos são indicados pela linha dupla

e são usados para especificar os valores de α, β e γ de acordo com a Tabela 1.1. Por exemplo, se o resultado da primeira medida for -1 e o da segunda $+1$, então $\alpha = 1$ e $\beta = \gamma = 0$. Como as medidas usadas no código de 3 qubits não mudam o estado do sistema quando o erro está no conjunto $\mathcal{E} = \{I, X_1, X_2, X_3\}$, podemos dar continuidade ao circuito após uma medição e aplicar as portas X_1^α, X_2^β e X_3^γ para corrigir o erro. A última etapa é a aplicação do operador de codificação inverso. Se o erro for algum operador do conjunto $\mathcal{E}$ ou uma combinação linear desses operadores, a saída do circuito será $|\psi\rangle|0\rangle|0\rangle$. Portanto, o estado original é recuperado.

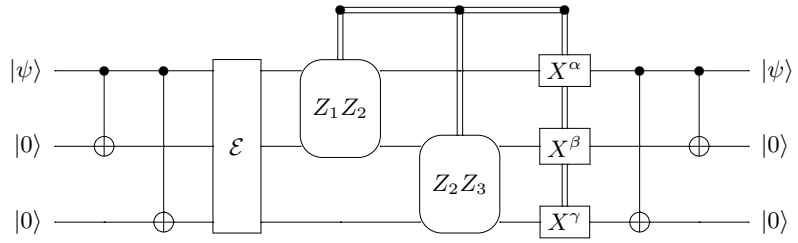

Figura 1.3: Circuito representando todo o processo de codificaçao, erro $\mathcal{E}$, síndrome e recuperação do código de 3 qubits. Os observáveis estão representados por caixas com ângulos arredondados. Os valores de α, β γ dependem dos resultados das medidas de acordo com a Tabela 1.1.

Toda medida intermediária pode ser colocada no final do circuito. A maneira mais simples de fazer essa conversão é usar qubits extras chamados de *ancillas*.

Exercício 1.1. *1. Verifique que o conjunto de mensagens possíveis (sem exigir normalização) que podem ser enviadas usando o código de 3 qubits é um espaço vetorial de dimensão 2 (denotado por $\mathcal{C}$) e é um subespaço do espaço de Hilbert $\mathcal{H}_{2^3}$.*

2. Mostre que o erro X_1 leva o conjunto de mensagens possíveis em um espaço vetorial de dimensão 2 ortogonal a $\mathcal{C}$. Verifique

também para X_2 e X_3 e, além disso, verifique que todos os subespaços envolvidos são ortogonais entre si.

3. *(Condições de Detecção de Erros) Seja $\mathcal{E} = \{I, X_1, X_2, X_3\}$ o conjunto dos possíveis erros do canal. Mostre que se $E \in \mathcal{E}$, existe c_E, que só depende do erro, tal que*

$$\langle i_L|E|j_L\rangle = c_E\,\delta_{ij},$$

para todo i, j. Quais são os valores de c_E para $E \in \mathcal{E}$?

4. *Os operadores lógicos $\bar{X}$ e $\bar{Z}$ são definidos de forma que $\bar{X}|0_L\rangle = |1_L\rangle$, $\bar{X}|1_L\rangle = |0_L\rangle$, $\bar{Z}|0_L\rangle = |0_L\rangle$ e $\bar{Z}|1_L\rangle = -|1_L\rangle$. Encontre uma expressão para esses operadores lógicos.*

5. *Mostre que se $\mathcal{E}$ satisfaz às condições de detecção de erros, então a combinação linear de erros em $\mathcal{E}$ também satisfaz.*

Exercício 1.2. *Faça um circuito equivalente ao da Fig. 1.3 sem medições intermediárias.*

1.3 Códigos Lineares Clássicos

Um código $\mathcal{C}$ linear binário (n, k) é obtido como imagem de uma *transformação linear injetiva* $\Phi : \mathbb{Z}_2^k \mapsto \mathbb{Z}_2^n$ tal que

$$\begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} \mapsto G_{n\times k} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix}.$$

$G_{n\times k}$ é a *matriz geradora* do código $\mathcal{C}$, que tem posto k. As colunas de G são linearmente independentes e geram o subespaço $\mathcal{C}$ de dimensão 2^k em $\mathbb{Z}_2^n$. Isto é, o código tem 2^k *palavras binárias* de comprimento n, que codificam as palavras originais que tinham k bits.

A matriz geradora não é única, pois podemos fazer uma mudança de base no subespaço $\mathcal{C}$. A apresentação canônica de G é da forma

$$G = \begin{bmatrix} I_{k\times k} \\ B_{(n-k)\times k} \end{bmatrix}. \tag{1.3.5}$$

Nesse caso, a codificação de uma palavra $(a_1, \cdots, a_k)$ é da forma

$$G_{n\times k}\cdot\begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} a_1 \\ \vdots \\ a_k \\ h_1 \\ \vdots \\ h_{n-k} \end{pmatrix},$$

onde $h_1, \cdots, h_{n-k}$ são combinações lineares das primeiras coordenadas. Em particular, a codificação da palavra $\mathbf{e}_j = (0, \cdots, 1, \cdots, 0)$ com um único 1 na j-ésima posição é a j-ésima coluna de G.

A *matriz verificadora* é da forma

$$H_{(n-k)\times n} = \begin{bmatrix} B_{(n-k)\times k} & I_{(n-k)\times(n-k)} \end{bmatrix}. \tag{1.3.6}$$

$H_{(n-k)\times n}$ é uma matriz capaz de detectar se um vetor $\mathbf{v} \in \mathbb{Z}_2^n$ é uma palavra do código ou não, pois

$$H\mathbf{v}^T = 0,$$

se, e somente se, $\mathbf{v}$ está no código. Isso se deve ao fato de que as colunas de G geram os vetores de $\mathcal{C}$ enquanto que as linhas de H geram os vetores do espaço ortogonal $\mathcal{C}^\perp$. Usando as Eqs. (1.3.5) e (1.3.6), podemos verificar que $H\,G = 0$. A matriz H serve também como análise de síndrome, como veremos na próxima seção em um caso particular.

1.3.1 Códigos de Hamming

Os *códigos de Hamming* $\mathcal{H}_r$ são *códigos lineares clássicos* do tipo $(2^r-1, 2^r-r-1, 3)$ para inteiros r positivos, portanto são subespaços de $\mathbb{Z}_2^{2^r-1}$. A *matriz de paridade* tem como colunas os elementos não nulos de $\mathbb{Z}_2^r$. Por exemplo, para $r = 3$

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

Escolhemos a ordem dos números de forma que H ficasse na forma canônica descrita na Eq. (1.3.6). Usando a Eq. (1.3.5) obtemos a seguinte matriz geradora:

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

Como G tem 4 colunas, o número total de palavras é 16. Fazendo todas as combinações lineares possíveis, obtemos

$$\begin{array}{cccc} 0000000 & 0001111 & 0010110 & 0011001 \\ 0100101 & 0101010 & 0110011 & 0111100 \\ 1000011 & 1001100 & 1010101 & 1011010 \\ 1100110 & 1101001 & 1110000 & 1111111. \end{array}$$

Tirando a palavra 0000000, todas as outras tem peso maior ou igual a 3, portanto a distância do código é 3.

Esse código é do tipo $(7, 4, 3)$. Como a distância é 3, ele consegue detectar erros em até duas letras e corrigir erros em até uma letra da palavra. Suponha que o erro tenha ocorrido no j-ésimo bit da palavra. Se a palavra do código era $\mathbf{v}$, ela passou a ser $\mathbf{v} + \mathbf{e}_j$. Aplicando H na palavra modificada e usando que $H\mathbf{v} = 0$, obtemos $H\mathbf{e}_j$. Como $H\mathbf{e}_j$ é a j-ésima coluna de H, por inspeção da matriz H, podemos determinar qual é a posição da coluna em questão e, portanto, determinar qual bit foi modificado.

Esse código tem uma propriedade interessante: para cada palavra do código, existem 7 *strings* de 7 bits que têm distância 1 da palavra selecionada. Todas essas *strings* não pertencem ao código. Esses 7 *strings* junto com a palavra do código formam uma bola de 8 elementos. Existem 16 bolas com interseção vazia 2-a-2 e que cuja união cobre todas as 2^7 *strings* possíveis. Um código com esta propriedade se chama *perfeito*.

Para qualquer código linear $\mathcal{C}$ com matriz geradora G, podemos definir um *código dual* $\mathcal{C}^{\perp}$, cuja matriz de paridade é $H^{\perp} = G^T$. As palavras do código dual são ortogonais a todas palavras do código original. Entretanto, pode acontecer de $C^{\perp} \subseteq C$. Nesse caso, o código é chamado de *auto-dual*. Os duais dos códigos de Hamming são códigos auto-duais. Vamos definir $H' = G^T$ e $G' = H^T$. Como $HG = 0$, segue que $H'G' = 0$. Portanto, a matriz geradora do código dual é H^T. Fazendo todas as combinações lineares das colunas, obtemos

$$\begin{array}{cccc} 0000000 & 0001111 & 0110011 & 0111100 \\ 1010101 & 1011010 & 1100110 & 1101001. \end{array}$$

O novo código é do tipo (7,3,4). Ele não é um *código perfeito*. Podemos verificar que todas as palavras de $\mathcal{C}^{\perp}$ estão em $\mathcal{C}$. Portanto, $C^{\perp} \subseteq C$, confirmando que o código dual ao código de Hamming é auto-dual.

1.4 Códigos Quânticos CSS

Os *códigos de Calderbank-Shor-Steane* (CSS) formam uma ampla classe de códigos quânticos que são construídos a partir dos *códigos lineares clássicos*. Eles pertencem a uma classe mais geral dos *códigos quânticos estabilizadores*. A estrutura desses códigos pode ser entendida através de um caso particular. Vamos tomar como base o código de Hamming (7,4,3) descrito na seção anterior. Os códigos CSS são construídos usando dois códigos lineares clássicos, $\mathcal{C}_1$ e $\mathcal{C}_2$, tal que $\mathcal{C}_2 \subseteq \mathcal{C}_1$. Vamos tomar $\mathcal{C}_1$ como o código de Hamming $(7, 4, 3)$ e $\mathcal{C}_2$ como o código dual $(7, 3, 4)$. O código quântico resultante será do tipo $[7, 1, 3]$.

Os códigos lineares clássicos têm a estrutura do espaço vetorial $\mathbb{Z}_2^n$. No contexto quântico, esse códigos são imersos em um espaço de Hilbert e passam a ser um espaço vetorial $\mathbb{C}^{2^n}$, correspondendo a n qubits. As palavras do código podem ser colocadas em superposição formando um estado quântico, que não poderia ser implementado diretamente em um computador clássico, porém é tratado eficientemente em um computador quântico. Em particular, vamos definir o estado quântico

$|\mathcal{C}_2\rangle$ da seguinte forma:

$$|\mathcal{C}_2\rangle = \frac{1}{\sqrt{|\ \mathcal{C}_2\ |}} \sum_{x \in \mathcal{C}_2} |x\rangle, \tag{1.4.7}$$

isto é, $|\mathcal{C}_2\rangle$ é a superposição quântica normalizada de todas as palavras do código $\mathcal{C}_2$.

Um código linear $\mathcal{C}$, além de ter a estrutura do espaço de Hilbert $\mathbb{C}^{2^n}$, tem também uma estrutura de grupo[1]. As palavras do código $\mathcal{C}$ formam um *grupo Abeliano* com a operação de soma binária bit-a-bit (XOR). O código $\mathcal{C}_2$ é portanto um subgrupo do código $\mathcal{C}_1$. O estado $|\mathcal{C}_2\rangle$ é a superposição de todos os elementos do subgrupo $\mathcal{C}_2$. Podemos agora definir a superposição de todos os elementos de uma *classe lateral* de $\mathcal{C}_2$ em $\mathcal{C}_1$. Seja $y \in \mathcal{C}_1$, vamos definir

$$|\mathcal{C}_2 + y\rangle = \frac{1}{\sqrt{|\ \mathcal{C}_2\ |}} \sum_{x \in \mathcal{C}_2} |x + y\rangle, \tag{1.4.8}$$

onde $C_2 + y$ é a classe lateral de $\mathcal{C}_2$ em $\mathcal{C}_1$ que contém y. Como todas as classes laterais tem a mesma cardinalidade, devemos usar a mesma normalização.

Os códigos CSS$(\mathcal{C}_1, \mathcal{C}_2)$ sobre os códigos lineares clássicos $\mathcal{C}_1$ e $\mathcal{C}_2$ do tipo (n, k_1) e (n, k_2), respectivamente, são códigos quânticos do tipo $[n, k_1 - k_2]$ gerados pelos estados quânticos das classes laterais de $\mathcal{C}_2$ em $\mathcal{C}_1$. O *índice* (número de classes laterais) de $\mathcal{C}_2$ em $\mathcal{C}_1$ é $|\mathcal{C}_1|/|\mathcal{C}_2|$, isto é, $2^{k_1-k_2}$. Portanto, o código CSS tem $2^{k_1-k_2}$ estados lógicos, que formam uma base ortonormal. Se T é uma transversal de $\mathcal{C}_2$ em $\mathcal{C}_1$, então a base do código CSS é $\{|\mathcal{C}_2 + y\rangle \,|\, y \in T\}$.

Exercício 1.3. *Se T é uma transversal de $\mathcal{C}_2$ em $\mathcal{C}_1$, mostre que $\{|\mathcal{C}_2 + y\rangle \,|\, y \in T\}$ é uma base ortonormal com $2^{k_1-k_2}$ elementos.*

Exercício 1.4. *Existe uma outra forma de descrever os códigos CSS. Sejam $\mathcal{C}_1$ e $\mathcal{C}_2$ códigos lineares clássicos do tipo (n, k_1) e (n, k_2), respectivamente, tal que $\mathcal{C}_2^{\perp} \subseteq \mathcal{C}_1$. As palavras do código são superposições*

[1]Veja o Apêndice A para uma breve revisão de alguns conceitos sobre a Teoria de Grupos.

das classes laterais de $\mathcal{C}_2^{\perp}$ *em* $\mathcal{C}_1$*. Mostre que nesse caso, o código CSS é do tipo* $(n, k_1 + k_2 - n)$*.*

1.4.1 Código de Steane

No exemplo onde $\mathcal{C}_1$ é o código de Hamming $(7, 4, 3)$ e $\mathcal{C}_2$ é o código dual $(7, 3, 4)$, o código quântico CSS é do tipo $[7, 1, 3]$, tem dimensão 2 e os estados lógicos são

$$\begin{aligned} |0_L\rangle &= \frac{1}{2\sqrt{2}}\big(|0000000\rangle + |0001111\rangle + |0110011\rangle + |0111100\rangle + \\ &\quad |1010101\rangle + |1011010\rangle + |1100110\rangle + |1101001\rangle\big) \end{aligned}$$

e

$$\begin{aligned} |1_L\rangle &= \frac{1}{2\sqrt{2}}\big(|1111111\rangle + |1110000\rangle + |1001100\rangle + |1000011\rangle + \\ &\quad |0101010\rangle + |0100101\rangle + |0011001\rangle + |0010110\rangle\big). \end{aligned}$$

O estado $|0_L\rangle$ foi obtido a partir das palavras do código de Hamming dual $(7, 3, 4)$ e o estado $|1_L\rangle$ foi obtido a partir da única classe lateral do código de Hamming dual no código de Hamming $(7, 4, 3)$. Esse código é conhecido como *código de Steane*. Ele detecta e corrige um erro genérico em 1 qubit. A análise de síndrome é feita através de uma medição em cascata dos seguintes 6 observáveis: $X_2X_3X_4X_5$, $X_1X_3X_4X_6$, $X_1X_2X_4X_7$, $Z_2Z_3Z_4Z_5$, $Z_1Z_3Z_4Z_6$, $Z_1Z_2Z_4Z_7$. A explicação do porquê esses observáveis fazem a análise de síndrome vem do uso do formalismo estabilizador, que será tratado na próxima seção. Porém, se olharmos a matriz de paridade H, podemos ver que o formato desses observáveis refletem as linhas de H. Sem o formalismo estabilizador, a análise do processo é extremamente trabalhosa e particular para esse código em questão. Para dar pelo menos uma intuição do porquê o processo funciona, vamos considerar um caso particular de erro: inversão de 1 qubit.

Uma palavra genérica enviada por Alice para Beto no canal quântico é da forma $|\psi\rangle = a|0\rangle + b|1\rangle$. A codificação é feita com um operador

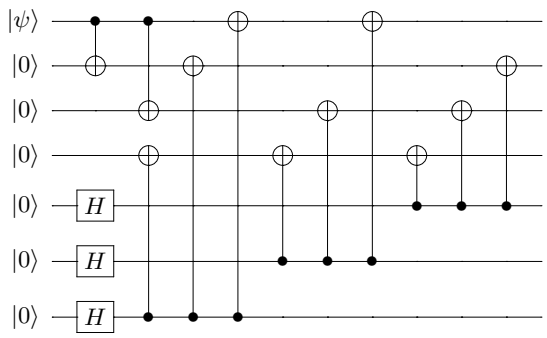

Figura 1.4: Circuito de codificação do código de Steane.

unitário que obedece

$$\begin{aligned} U_{\rm cod}|0000000\rangle &= |0_L\rangle, \\ U_{\rm cod}|1000000\rangle &= |1_L\rangle. \end{aligned}$$

O operador $U_{\rm cod}$ está especificado na Fig. 1.4 em termos das portas elementares universais H e CNOT. Após a codificação, o estado $|\psi_{\rm cod}\rangle = a|0_L\rangle + b|1_L\rangle$ é enviado pelo canal quântico. Se ocorrer a inversão de qubit no primeiro qubit, Beto vai receber o estado $|\psi'\rangle = aX_1|0_L\rangle + bX_1|1_L\rangle$, onde

$$\begin{aligned} X_1|0_L\rangle &= \frac{1}{2\sqrt{2}}\big(|1000000\rangle + |1001111\rangle + |1110011\rangle + |1111100\rangle + \\ &\quad |0010101\rangle + |0011010\rangle + |0100110\rangle + |0101001\rangle\big) \end{aligned}$$

e

$$\begin{aligned} X_1|1_L\rangle &= \frac{1}{2\sqrt{2}}\big(|0111111\rangle + |0110000\rangle + |0001100\rangle + |0000011\rangle + \\ &\quad |1101010\rangle + |1100101\rangle + |1011001\rangle + |1010110\rangle\big). \end{aligned}$$

O estado $|\psi'\rangle$ é ortogonal ao código quântico, pois ele é ortogonal tanto a $|0_L\rangle$ como a $|1_L\rangle$. Além disso, dois erros em qubits distintos,

por exemplo $X_1|\psi_{\text{cod}}\rangle$ e $X_2|\psi_{\text{cod}}\rangle$, também estão associados a espaços ortogonais entre si. Quando erros distintos são levados em espaços ortogonais, se diz que os erros são distinguíveis. O que temos que fazer é o seguinte: escolhemos um observável cuja decomposição em projetores coincide exatamente com os projetores nesses espaços ortogonais. Uma medida desse observável vai permitir Beto identificar se o estado recebido está com problemas, isto é, se é ortogonal ao código e mais que isso, vai permitir identificar para qual dos espaços ortogonais ao código a mensagem corrompida foi enviada. A partir dessa informação, Beto pode aplicar a correção no qubit correto. Note que há espaço de sobra ortogonal ao código quântico para todos possíveis erros de inversão em 1 qubit. A análise completa de todas as possibilidades será feita na próxima seção usando um formalismo bem mais poderoso, chamado formalismo estabilizador.

Exercício 1.5. *1. Seja $\mathcal{C}$ o subespaço vetorial gerado pelos estados lógicos $|0_L\rangle$ e $|1_L\rangle$ do código de Steane. Mostre que o erro Z_1 leva o conjunto de mensagens possíveis em um espaço vetorial de dimensão 2 ortogonal a $\mathcal{C}$. Verifique também para Z_2 até Z_7 e, além disso, verifique que todos os subespaços envolvidos são ortogonais entre si.*

2. (Condições de Detecção de Erros) Seja $\mathcal{E} = \{I, X_1, \cdots, X_7, Z_1, \cdots, Z_7\}$ o conjunto dos possíveis erros do canal. Mostre que se $E \in \mathcal{E}$, existe c_E, que só depende do erro, tal que

$$\langle i_L|E|j_L\rangle = c_E\,\delta_{ij},$$

para todo i, j. Quais são os valores de c_E para $E \in \mathcal{E}$?

3. Os operadores lógicos $\bar{X}$ e $\bar{Z}$ são definidos de forma que $\bar{X}|0_L\rangle = |1_L\rangle$, $\bar{X}|1_L\rangle = |0_L\rangle$, $\bar{Z}|0_L\rangle = |0_L\rangle$ e $\bar{Z}|1_L\rangle = -|1_L\rangle$. Encontre uma expressão para esses operadores lógicos.

Sugestões para Leitura

Uma excelente referência para uma introdução aos códigos clássicos é [17]. Para uma rápida introdução aos códigos quânticos, sugerimos a Ref. [15]. A Ref. [20] também é muito útil. Os primeiros trabalhos que apareceram na literatura sobre códigos quânticos foram [23] e [26]. Os códigos CSS foram introduzidos nas Refs. [4] e [27].

Capítulo 2

Códigos Aditivos

Os *códigos quânticos aditivos*, generalização dos *códigos clássicos lineares*, podem ser totalmente expressos em termos do *formalismo estabilizador*. O formalismo estabilizador é um método de expressar estados quânticos por um conjunto de operadores do *grupo de Pauli*. Tanto a descrição da *evolução quântica* como da *medida quântica* devem ser adaptadas a esse formalismo. No entanto, o método não é totalmente geral, pois nem todos os *estados quânticos* podem ser expressos em termos do formalismo estabilizador.

Vamos destacar alguns fatos que serão importantes para a aplicação do formalismo estabilizador para a descrição dos códigos estabilizadores. As provas para esses fatos podem ser encontradas nas referências ao final deste livro.

2.1 Formalismo Estabilizador

Vamos começar descrevendo o formalismo estabilizador com um exemplo. Considere o seguinte estado da *base de Bell*:

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

Esse estado satisfaz

$$\begin{aligned} X_1X_2|\psi\rangle &= |\psi\rangle, \\ Z_1Z_2|\psi\rangle &= |\psi\rangle. \end{aligned}$$

Nesse caso, diz-se que o estado $|\psi\rangle$ é *estabilizado* tanto pelo operador $X_1X_2 = X \otimes X$ como por $Z_1Z_2 = Z \otimes Z$. A menos de uma fase global, o estado $|\psi\rangle$ é o único *estado estabilizado* pelos operadores X_1X_2 e Z_1Z_2. A ideia do formalismo é descrever o estado $|\psi\rangle$ pelo conjunto $\{X_1X_2,\ Z_1Z_2\}$. Esta opção pode parecer estranha a primeira vista, porém veremos que há uma enorme economia na descrição do sistema físico e da sua evolução.

O poder do formalismo estabilizador reside em alguns fatos básicos da Teoria de Grupos. Em particular, se um grupo G tem $|G|$ elementos, então é possível achar um *conjunto gerador* com no máximo $\lceil \log_2 |G| \rceil$ elementos. No caso do formalismo estabilizador, o grupo em questão é o *grupo de Pauli* G_n, onde n é o número de qubits. Para um qubit, o grupo de Pauli é

$$G_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{2.1.1}$$

É fácil de verificar que este conjunto satisfaz as propriedades de fechamento, existência de elemento neutro ($I_{2\times 2}$), associatividade e existência de elemento inverso (próprio elemento ou o negativo do elemento) com relação ao produto usual de matrizes. Quando passamos para 2 qubits, o grupo G_2 é obtido tomando o produto tensorial dos elementos de G_1, isto é

$$G_2 = \{\pm I\otimes I, \pm iI\otimes I, \pm I\otimes X, \pm iI\otimes X, \pm I\otimes Y, \cdots, \pm iZ\otimes Z\}. \tag{2.1.2}$$

No caso geral de n qubits, os elementos do grupo G_n são produtos tensoriais de n matrizes de Pauli com fatores multiplicativos ± 1 e $\pm i$. Daqui para frente vamos tomar n como o número de qubits em questão.

Um conjunto estabilizador, que vamos denotar por S, é um subgrupo comutativo do grupo de Pauli G_n que não contmém $-I$. O conjunto S tem um espaço vetorial associado, denotado por V_S. V_S

consiste dos vetores do espaço de Hilbert $\mathcal{H}$ de n qubits que são estabilizados por todos elementos de S. V_S é um subespaço de $\mathcal{H}$. Esta definição é coerente pois se $|\psi_1\rangle$ e $|\psi_2\rangle$ são estabilizados por S, então uma combinação linear de $|\psi_1\rangle$ e $|\psi_2\rangle$ também é estabilizada por S.

Fato 1

Se $-I \in S$, então V_S é o espaço vetorial trivial gerado pelo vetor nulo.

O Fato 1 é verdadeiro porque o único vetor estabilizado por $-I$ (um vetor $|\psi\rangle$ tal que $|\psi\rangle = -|\psi\rangle$) é o vetor nulo. Por isso assumimos que $-I \notin S$. Segue como consequência que $\pm iI \notin S$.

Fato 2

Seja S um subgrupo não-comutativo de G_n. Então, V_S é o espaço vetorial trivial gerado pelo vetor nulo.

Uma segunda exigência para que V_S não seja o espaço vetorial nulo é que S seja um subgrupo comutativo. As matrizes de Pauli obedecem às seguintes propriedades: duas matrizes de Pauli comutam ou anticomutam. Não existe outra opção. Se S não for um grupo comutativo, deve existir g_1 e g_2 tal que $g_1 g_2 = -g_2 g_1$. Usando este fato podemos mostrar que se $g_1, g_2 \in S$ e $g_1 g_2 = -g_2 g_1$ então $g_1 g_2 |\psi\rangle = -g_2 g_1 |\psi\rangle$ e, portanto, $|\psi\rangle = -|\psi\rangle$. Por isso assumimos que S seja comutativo.

Fato 3

Seja $S = \langle g_1, \cdots, g_{n-k} \rangle$, para algum $0 \leq k < n$, isto é, S é gerado por $n-k$ elementos do grupo de Pauli G_n. Então, V_S é um espaço vetorial de dimensão 2^k.

A demonstração do Fato 3 é trabalhosa. Ela pode ser encontrada nas referências. A compreensão desse fato é fundamental para ve-

rificarmos a consistência do que se segue. Note que, quanto menos elementos tiver o conjunto gerador de S, maior será a dimensão do espaço vetorial V_S. Para representar um estado por S, o espaço vetorial V_S deve ter dimensão 1. Portanto, o conjunto gerador de S deve ter n elementos.

Uma outra maneira de visualizar V_S é pela interseção dos espaços vetoriais associados a cada elemento do conjunto gerador de S. Isto é, tome o primeiro elemento g_1 e descubra qual é o espaço estabilizado por este elemento. A dimensão do espaço vetorial V_{g_1} é 2^{n-1}. Faça isso para os outros elementos, e tome a interseção:

$$V_S = V_{g_1} \cap \cdots \cap V_{g_{n-k}}. \tag{2.1.3}$$

Por exemplo, para $n = 3$, tome $S = \langle g_1, g_2 \rangle$ onde $g_1 = Z_1 Z_2$, $g_2 = Z_2 Z_3$. Portanto, $S = \{I_{8\times 8}, Z_1 Z_2, Z_2 Z_3, Z_1 Z_3\}$. Vamos determinar V_{g_1}. Por inspeção, descobrimos que $|000\rangle$, $|001\rangle$, $|110\rangle$ e $|111\rangle$ são estabilizados por g_1. Pelo Fato 2, V_{g_1} tem dimensão 4, portanto acabamos de determinar uma base para esse espaço vetorial. Analogamente, os vetores $|000\rangle$, $|011\rangle$, $|100\rangle$ e $|111\rangle$ formam uma base para V_{g_2}. Tomando a interseção, obtemos que $|000\rangle$ e $|111\rangle$ formam uma base para V_S.

Exercício 2.1. *Ache geradores dos subgrupos estabilizadores para cada um dos estados de Bell:*

$$\frac{|00\rangle \pm |11\rangle}{\sqrt{2}}, \frac{|01\rangle \pm |10\rangle}{\sqrt{2}}.$$

Exercício 2.2. *Ache geradores para o subgrupo estabilizador dos estados de Greenberger-Horne-Zeilinger (GHZ) de n qubits*

$$\frac{|0\rangle^{\otimes n} + |1\rangle^{\otimes n}}{\sqrt{2}}.$$

2.2 Portas Unitárias no Formalismo Estabilizador

Quando descrevemos espaços vetoriais usando o formalismo estabilizador, estamos lidando com o postulado do espaço de estados da

Mecânica Quântica. O próximo passo é descrever as operações dinâmicas. O postulado da evolução dinâmica afirma que se no instante inicial o sistema físico é descrito pelo estado $|\psi\rangle$, então, após a evolução, o estado será descrito por $U|\psi\rangle$, para algum operador unitário U, que depende dos processos físicos em questão. No formalismo estabilizador, representamos os *estados quânticos* por subgrupos S do *grupo de Pauli* G_n. Após a evolução, temos que descrever o sistema físico por um novo subgrupo do grupo de Pauli. A questão é como determinar esse novo subgrupo conhecendo-se S e U.

Suponha que aplicamos um operador unitário U sobre um espaço vetorial V_S que é estabilizado pelo grupo S. Seja $|\psi\rangle$ um estado de V_S. Então para qualquer elemento $g \in S$

$$U|\psi\rangle = Ug|\psi\rangle = UgU^\dagger U|\psi\rangle. \tag{2.2.4}$$

Isso mostra que $U|\psi\rangle$ é estabilizado por $UgU^\dagger$. Portanto, se $|\psi\rangle$ é estabilizado por S, $U|\psi\rangle$ é estabilizado por $USU^\dagger := \{UgU^\dagger,\ g \in S\}$. Além disso, se $g_1, \cdots, g_k$ geram S, então $Ug_1U^\dagger, \cdots, Ug_kU^\dagger$ geram $USU^\dagger$. Estas observações mostram completamente como caracterizar a evolução unitária no formalismo estabilizador. Se no instante inicial, o estado quântico é descrito por S, então no instante posterior, após a evolução, o estado será descrito por $USU^\dagger$, que é uma *operação de conjugação* de cada elemento de S.

Dado um *grupo estabilizador* no instante inicial, sabemos como calcular o grupo estabilizador após a evolução descrita por um operador unitário genérico U. Porém, pode ocorrer um problema nesse ponto. O formalismo estabilizador não é totalmente genérico. Ele não pode substituir completamente o formalismo usual baseado em vetores do espaço de Hilbert, pois existem estados que não são estabilizados por nenhum elemento do grupo de Pauli, a não ser pela identidade. Por exemplo, o estado

$$|\psi\rangle = \frac{|0\rangle + \mathrm{e}^{i\frac{\pi}{4}}|1\rangle}{\sqrt{2}} \tag{2.2.5}$$

não é estabilizado por nenhuma das matrizes do grupo G_1 na Eq. (2.1.1), exceto pela identidade. Portanto, para usar o formalismo estabilizador, temos que iniciar o processo com um estado que pode ser esta-

bilizado por algum subgrupo S. A princípio isso não é uma restrição séria, pois podemos iniciar o processo com um estado da base computacional e depois gerar estados mais complexos através de operadores unitários. No entanto, se o operador U gerar um estado que não pode ser estabilizado por nenhum elemento do grupo de Pauli, o formalismo estabilizador não pode ser aplicado. Quais operadores U podem ser usados no formalismo estabilizador? Uma boa forma de responder isso e verificar quais operadores universais podem ser usados. O *conjunto universal* mais usado é formado pelo operador CNOT de 2 qubits e pelos operadores de 1 qubit

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \quad \text{e} \quad T = \begin{bmatrix} 1 & 0 \\ 0 & \mathrm{e}^{i\frac{\pi}{4}} \end{bmatrix},$$

conhecidos como *Hadamard*, *fase* e *porta* T, respectivamente.

Vamos começar com o operador H. Um cálculo direto mostra que

$$HXH^\dagger = Z, \quad HYH^\dagger = -Y, \quad HZH^\dagger = X.$$

Isso é uma excelente notícia. Se o estado inicial é estabilizado por elementos do grupo de Pauli, as relações acima mostram que, após a aplicação do operador Hadamard, o novo estado vai continuar sendo estabilizado por elementos do grupo de Pauli. A regra é trocar X por Z, vice-versa e trocar Y por $-Y$ para cada qubit que tenha sido transformado pelo operador H.

Agora vamos analisar o operador S. Um cálculo direto mostra que

$$SXS^\dagger = Y, \quad SZS^\dagger = Z.$$

Multiplicando as relações acima e usando que $XZ = -iY$ obtemos

$$SYS^\dagger = -X.$$

Essas relações mostram que não há problema algum com a evolução governada pelo operador S.

O operador CNOT admite diversas possibilidades. Todas elas podem ser obtidas das seguinte relações:

$$UX_1U^\dagger = X_1X_2, \quad UX_2U^\dagger = X_2, \quad UZ_1U^\dagger = Z_1, \quad UZ_2U^\dagger = Z_1Z_2$$

onde $U = \text{CNOT}$. Também não há problemas com relação ao CNOT.

A última porta universal gera um problema incontornável, pois

$$TXT^\dagger = \frac{X+Y}{\sqrt{2}}.$$

Portanto, a evolução produzida pela porta T gera estados que não são estabilizados pelos elementos do grupo de Pauli. Eles são estabilizados por um operador que é a soma de matrizes de Pauli. Não pode.

Concluímos que qualquer operador obtido por multiplicação das portas CNOT, H e S e produto tensorial entre elas está incorporado ao formalismo estabilizador. Isso quer dizer que se o estado inicial é estabilizado por um conjunto de elementos do grupo de Pauli, após a evolução produzida por esse subconjunto de portas universais, o estado será estabilizado por um outro conjunto de elementos do grupo de Pauli.

O formalismo estabilizador produz uma enorme economia na notação de estados quânticos. Um estado genérico de n qubits é descrito na base computacional por 2^n coeficientes complexos. Se esse estado é estabilizado por um subgrupo S do grupo de Pauli G_n, então podemos caracterizá-lo usando um conjunto gerador para S, cujo número de elementos será $O(n)$. Também haverá economia de notação na descrição da evolução desse estado, pois temos que operar o operador de evolução por conjugação sobre $O(n)$ elementos. Falta ainda descrever a etapa final, que é a realização de uma *medida física*. Até o momento, estamos perto de mostrar que qualquer circuito que usa as portas CNOT, H e S pode ser simulado eficientemente por um computador clássico. Toda a riqueza da computação quântica, frente a computação clássica, fica a cargo da porta T. Esse impressionante resultado é conhecido como *teorema de Gottesman-Knill*.

2.3 Medida no Formalismo Estabilizador

Para completar o programa de descrever os postulados da Mecânica Quântica em termos do formalismo estabalizador para um conjunto particular de operadores unitários, temos que descrever o *processo da*

medida. Em particular, vamos mostrar que medidas usando matrizes de Pauli ou produto tensorial de matrizes de Pauli como *observável* preservam o formalismo estabilizador e, portanto, a *medida na base computacional* também preserva. Suponha que o sistema físico seja descrito pelo estado $|\psi\rangle$, que é estabilizado por $S = \langle g_1, \cdots, g_n\rangle$. Tome agora como observável g, um produto tensorial de n matrizes de Pauli. Esse observável é um elemento do grupo de Pauli G_n. Queremos determinar como S se transforma após a medida do observável g. Temos duas possibilidades:

1) g comuta como todos os geradores de S, ou

2) g anti-comuta com pelo menos um dos geradores de S.

Vamos analisar a possibilidade 1). Se g comuta com todos geradores de S, então $g_j g|\psi\rangle = g g_j|\psi\rangle = g|\psi\rangle$, para todos os geradores g_j, isto é, $g|\psi\rangle$ é estabilizado por S. Segue que $g|\psi\rangle$ pertence a V_S e $g|\psi\rangle$ tem que ser um múltiplo de $|\psi\rangle$, pois V_S tem dimensão 1. Uma vez que $g^2 = I$, temos duas possibilidades: $g|\psi\rangle = \pm|\psi\rangle$. Portanto, $g \in S$ ou $-g \in S$. Os possíveis resultados da medida são ± 1, pois esses são os autovalores das matrizes de Pauli. Se $g \in S$, a média dos possíveis resultados da medida com g é $\langle g\rangle = \langle\psi|g|\psi\rangle = 1$. Isso quer dizer que o resultado da medida nunca é -1, portanto será sempre igual a 1. Se $-g \in S$, então $\langle g\rangle = -1$ e o resultado da medida será sempre igual a -1. O estado após a medida pode ser obtido usando os seguintes fatos: $g = P_+ - P_-$ e $I = P_+ + P_-$, onde P_+ é o projetor no auto-espaço associado ao $+1$ e P_- é o projetor associado ao -1. Portanto,

$$P_+ = \frac{I+g}{2}, \tag{2.3.6}$$

$$P_- = \frac{I-g}{2}. \tag{2.3.7}$$

A partir destas expressões, podemos verificar que o estado $|\psi\rangle$ fica inalterado após a medida.

Vamos analisar a possibilidade 2). Se g não comuta com um dos geradores de S, basta considerar o seguinte caso: g não comuta apenas com g_1, pois se g também anti-comuta com g_2, podemos substituir g_2 por $g_1 g_2$ e assim sucessivamente com todos os geradores que anti-comutam com g. No final, teremos um novo conjunto gerador para

S com a propriedade desejada. Os projetores associados à medida do observável g estão descritos nas Eqs. (2.3.6) e (2.3.7). No entanto, o estado $|\psi\rangle$ não é estabilizado por S. As probabilidades associadas a cada autovalor são

$$p_{\pm} = \langle\psi|P_{\pm}|\psi\rangle\,. \tag{2.3.8}$$

Usando o fato $\langle\psi|g|\psi\rangle = \langle\psi|gg_1|\psi\rangle = -\langle\psi|g_1g|\psi\rangle$ temos que $\langle\psi|g|\psi\rangle = 0$. Usando as Eqs. (2.3.6) e (2.3.7) na Eq. (2.3.8), obtemos $p_+ = p_-$. Como $p_+ + p_- = 1$, segue que $p_+ = p_- = 1/2$. Se o resultado da medida for $+1$, o estado do sistema logo após será

$$|\psi'\rangle = \frac{1}{\sqrt{2}}(I+g)|\psi\rangle,$$

que é estabilizado por $S' = \langle g, g_2, \cdots, g_n\rangle$. Se o resultado da medida for -1, o estado do sistema logo após será

$$|\psi'\rangle = \frac{1}{\sqrt{2}}(I-g)|\psi\rangle,$$

que é estabilizado por $S' = \langle -g, g_2, \cdots, g_n\rangle$.

A *medida na base computacional* é realizada com *medidas em cascata* usando o operador Z, um qubit de cada vez. Os resultados acima se aplicam nesse caso. Portanto, podemos acompanhar os resultados intermediários usando o formalismo estabilizador.

2.4 Código de Shor

O *código de Shor* é um código $[9, 1, 3]$, onde $S = \langle g_1, \cdots, g_8\rangle$, de acordo com a Tabela 2.1. Pelo Fato 3, concluímos que V_S é um subespaço de dimensão 2, representando 1 *qubit lógico*, do espaço de Hilbert de dimensão 2^9. Nosso objetivo agora é achar uma base para V_S.

O exemplo no final da Sec. 2.1 mostra que qualquer produto tensorial dos vetores $|000\rangle$ e $|111\rangle$ entre si com 3 termos, de forma a representar uma estado de 9 qubits, por exemplo $|000\rangle|000\rangle|111\rangle$, é estabilizado por $g_1, \cdots, g_6$. Como há 8 possibilidades de vetores independentes, temos um espaço estabilizado de dimensão 8. Qualquer

Gerador	Expressão
g_1	Z_1Z_2
g_2	Z_2Z_3
g_3	Z_4Z_5
g_4	Z_5Z_6
g_5	Z_7Z_8
g_6	Z_8Z_9
g_7	$X_1X_2X_3X_4X_5X_6$
g_8	$X_4X_5X_6X_7X_8X_9$
$\bar{X}$	$Z\,Z\,Z\,Z\,Z\,Z\,Z\,Z\,Z$
$\bar{Z}$	$X\,X\,X\,X\,X\,X\,X\,X\,X$

Tabela 2.1: Geradores do código de Shor e os operadores lógicos $\bar{X}$ e $\bar{Z}$.

combinação linear destes vetores da base também são estabilizados. Falta tratar g_7 e g_8. Um operador do tipo $X_1X_2X_3$ converte $|000\rangle$ em $|111\rangle$ e vice-versa. Portanto, o estado de 3 qubits

$$|\psi_+\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}$$

é estabilizado por $X_1X_2X_3$. Note que o estado

$$|\psi_-\rangle = \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

é não estabilizado por $X_1X_2X_3$, pois há uma troca de sinal. No entanto, $|\psi_-\rangle|\psi_-\rangle$ é estabilizado por $X_1X_2X_3X_4X_5X_6$. Analisando todos os produtos tensoriais de $|\psi_+\rangle$ e $|\psi_-\rangle$ que formam um estado de 9 qubits, concluímos que, para ser estabilizado tanto por g_7 como por g_8, as únicas possibilidades são

$$|0_L\rangle = |\psi_+\rangle|\psi_+\rangle|\psi_+\rangle, \tag{2.4.9}$$

$$|1_L\rangle = |\psi_-\rangle|\psi_-\rangle|\psi_-\rangle. \tag{2.4.10}$$

Os estados $|0_L\rangle$ e $|1_L\rangle$ são estabilizados por S e, portanto, formam uma base ortonormal para V_S. Esses estados representam as *palavras lógicas* do código, isto é, os qubits $|0\rangle$ e $|1\rangle$ que eram usados antes da *codificação* para escrever uma mensagem devem ser substituídos por $|0_L\rangle$ e $|1_L\rangle$. Se a mensagem for uma *string* de n qubits, a mensagem codificada terá $9n$ qubits. No espaço original, os operadores X e Z são muito úteis. X converte $|0\rangle$ em $|1\rangle$ e vice-versa. Z mantém $|0\rangle$ inalterado e inverte o sinal de $|1\rangle$. No espaço codificado, os operadores $\bar{X}$ e $\bar{Z}$, descritos na Tabela 2.1, fazem o papel dos operadores X e Z. Note que tanto $\bar{X}$ como $\bar{Z}$ comutam com todos os geradores de S. O estado $|0_L\rangle$ é estabilizado por $S_0 = \langle g_1, \cdots, g_8, \bar{Z}\rangle$ enquanto que $|1_L\rangle$ é estabilizado por $S_1 = \langle g_1, \cdots, g_8, -\bar{Z}\rangle$. Essa é uma maneira padrão de determinar os *estados lógicos*. No caso geral, quando V_S tem dimensão 2^k, teremos k *operadores lógicos* do tipo $\bar{Z}$ independentes, gerando um número maior (2^k) de estados lógicos.

Como o código de Shor tem distância 3, ele corrige um erro genérico em 1 qubit. Um erro genérico E sobre um qubit pode ser expresso como uma combinação linear de matrizes de Pauli

$$E = aI + bX + cY + dZ. \tag{2.4.11}$$

Nesse ponto precisamos do seguinte fato:

Fato 4

Um código que detecta e corrige erros produzidos por um conjunto de operadores, também detecta e corrige erros produzidos pela combinação linear desses operadores.

Esse fato é fundamental no que se segue, porém sua prova é trabalhosa, de forma que os detalhes podem ser obtidos nas referências. Portanto, se formos capazes de detectar e corrigir erros provocados por matrizes de Pauli sobre um único qubit, seremos capazes de corrigir um erro genérico em um qubit. Daqui para frente, vamos supor então que o erro foi produzido por uma das matrizes de Pauli em um único qubit.

O primeiro passo é a *codificação*. Suponha que Alice vai enviar

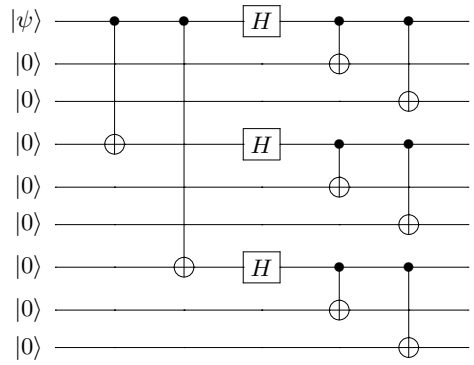

Figura 2.1: Circuito de codificação do código de Shor.

o estado $|\psi_0\rangle = a|0\rangle + b|1\rangle$ pelo canal quântico. Adicionamos 8 qubits auxiliares todos no estado $|0\rangle$. A codificação deve transformar simultaneamente

$$\begin{aligned} |000000000\rangle &\rightarrow |0_L\rangle, \\ |100000000\rangle &\rightarrow |1_L\rangle. \end{aligned}$$

Essa codificação é feita com o circuito da Fig. 2.1. A apresentação do operador unitário de codificação na forma de circuito é muito interessante quando o circuito é descrito em termos das portas universais ou em termos de portas cuja decomposição em portas universais é conhecida. Dessa forma fica imediatamente evidente se o operador de codificação é unitário e fica fácil de verificar se ele é eficientemente implementável.

O segundo passo, após o envio da mensagem pelo canal, é a *análise de síndrome*. No formalismo estabilizador, isso é feito fazendo-se medidas dos geradores de S, isto é, os observáveis são os geradores. Se não tiver ocorrido erro, ou equivalentemente, se o erro foi produzido por I, a medida em cascata, primeiro g_1 depois g_2 e assim sucessivamente, vai resultar na sequência $\{1, \cdots, 1\}$, com n uns, e o estado do sistema fica inalterado, pois todas as medidas recaem na possibilidade 1) analisada na Sec. 2.3.

Vamos supor que o erro ocorreu no primeiro qubit e foi produzido pela matriz X. Se estado original do sistema após a codificação era $|\psi\rangle$, após a ocorrência do erro ele foi modificado para $X_1|\psi\rangle$. Pela Tabela 2.1, podemos verificar que X_1 não pertence a S. De fato, X_1 anti-comuta com g_1 e como S é um grupo comutativo, segue que $X_1 \notin S$. Note que X_1 comuta com todo os outros geradores. Vamos mostrar agora que o resultado da medida do observável g_1 será -1 e continuará sendo 1 quando a medida usar os outros geradores como observáveis. Observe que os projetores têm a forma descrita nas Eqs. (2.3.6) e (2.3.7), onde g é o observável em questão. Portanto, a probabilidade do resultado ser -1 é

$$\begin{aligned} p_- &= \langle\psi|X_1^\dagger P_- X_1|\psi\rangle \\ &= \langle\psi|P_+|\psi\rangle \\ &= 1. \end{aligned}$$

Para mostrar que $X_1^\dagger P_- X_1 = P_+$, usamos a Eq. (2.3.7) (substituindo g por g_1), $g_1 X = -X g_1$ e $X^\dagger X = I$. Para mostrar que $\langle\psi|P_+|\psi\rangle = 1$, usamos a Eq. (2.3.6) e o fato que g_1 estabiliza $|\psi\rangle$. Como $p_- = 1$, segue que $p_+ = 0$. O cálculo análogo usando os outros geradores que comutam com X_1 mostra que $p_+ = 1$ em todos os casos. A medida em cascata vai resultar na sequência $\{-1, 1, 1, 1, 1, 1, 1, 1\}$ e o estado do sistema fica inalterado. A correção nesse caso deve ser a aplicação de $X_1^\dagger$.

Vamos supor que o erro ocorreu no primeiro qubit e foi produzido pela matriz Z. Se estado original do sistema era $|\psi\rangle$, após a ocorrência do erro ele foi modificado para $Z_1|\psi\rangle$. Pela Tabela 2.1, podemos verificar que Z_1 não pertence a S. De fato, Z_1 anti-comuta com g_7 e comuta com todos os outros geradores. Portanto, o resultado da medida será -1 para o observável g_7 e continuará sendo 1 quando a medida usar os outros geradores como observáveis. A medida em cascata vai resultar na sequência $\{1, 1, 1, 1, 1, 1, -1, 1\}$ e o estado do sistema fica inalterado. A correção nesse caso deve ser a aplicação de $Z_1^\dagger$. Note que se o erro tivesse sido Z_2 ou Z_3, a análise de síndrome teria dado o mesmo resultado e a aplicação de Z_1 teria corrigido da mesma forma, pois o efeito dos operadores Z_1, Z_2 e Z_3 nos estados $|\psi_+\rangle$ e $|\psi_-\rangle$ é o mesmo.

Vamos supor que o erro ocorreu no primeiro qubit e foi produzido pela matriz Y. Se estado original do sistema era $|\psi\rangle$, após a ocorrência do erro ele foi modificado para $Y_1|\psi\rangle$. Pela Tabela 2.1, podemos verificar que Y_1 não pertence a S. De fato, Y_1 anti-comuta com g_1 e com g_7 e comuta com todos os outros geradores. Portanto, o resultado da medida será -1 para os observáveis g_1 e g_7 e continuará sendo 1 quando a medida usar os outros geradores como observáveis. A medida em cascata vai resultar na sequência $\{-1, 1, 1, 1, 1, 1, -1, 1\}$ e o estado do sistema fica inalterado. A correção nesse caso deve ser a aplicação de Y_1. Esse resultado também pode ser obtido a partir dos resultados com os observáveis X e Z.

Erro	Resultados das medidas	Correção
I	1, 1, 1, 1, 1, 1, 1, 1	I
X_1	-1, 1, 1, 1, 1, 1, 1, 1	X_1
X_2	-1,-1, 1, 1, 1, 1, 1, 1	X_2
X_3	1,-1, 1, 1, 1, 1, 1, 1	X_3
Z_1 ou Z_2 ou Z_3	1, 1, 1, 1, 1, 1,-1, 1	Z_1
X_4	1, 1,-1, 1, 1, 1, 1, 1	X_4
X_5	1, 1,-1,-1, 1, 1, 1, 1	X_5
X_6	1, 1, 1,-1, 1, 1, 1, 1	X_6
Z_4 ou Z_5 ou Z_6	1, 1, 1, 1, 1, 1,-1,-1	Z_4
X_7	1, 1, 1, 1,-1, 1, 1, 1	X_7
X_8	1, 1, 1, 1,-1,-1, 1, 1	X_8
X_9	1, 1, 1, 1, 1,-1, 1, 1	X_9
Z_7 ou Z_8 ou Z_9	1, 1, 1, 1, 1, 1, 1,-1	Z_7

Tabela 2.2: Possíveis erros de 1 qubit no código de Shor, análise de síndrome e os operadores de correção. Os erros produzidos pela matriz de Pauli Y podem ser analisados através do produto dos erros X por Z. Os resultados das medidas são obtidos da seguinte forma: $+1$ se o erro comuta com o gerador do código e -1 se o erro anticomuta.

A Tabela 2.2 resume todos os possíveis erros de 1 qubit, mostra os resultados da análise de síndrome e os operadores de correção em cada caso. Os erros produzidos pela matriz de Pauli Y podem ser analisados através do produto dos erros X por Z.

Vamos dar um exemplo de um erro que não pode ser corrigido: $X_1X_2X_3$. Esse operador comuta com todos os geradores do código de Shor e a análise de síndrome retornará a sequência $\{1,1,1,1,1,1,1,1\}$ indicando que deveríamos corrigir com o operador I. Qualquer erro que seja um operador do grupo de Pauli $\mathcal{G}_n$ que não está em S, mas que comuta com todos os geradores de S, não pode ser corrigido. Portanto, a distância do código é o menor peso entre todos os operadores de $\mathcal{G}_n$ que estão no normalizador de S mas não estão em S. Mostramos que nenhum operador de um qubit está nessa classe, portanto podem ser corrigidos. Como o código tem distância 3, os erros que são operadores de 2 qubits podem ser detectados, porém não podem ser corrigidos. Por exemplo, o erro X_1X_2 tem a análise de síndrome igual ao erro X_3. Portanto, verificamos que houve erro, porém a correção sugerida na Tabela 2.2 não funciona.

2.5 Código Quântico [[5,1,3]]

O código quântico $[[5,1,3]]$ é a menor *codificação* possível de um qubit com distância 3. Esse código satura o *limite quântico de Hamming*. Os geradores do grupo estabilizador estão descritos na Tabela 2.3.

Gerador	Expressão
g_1	$X_1Z_2Z_3X_4$
g_2	$X_2Z_3Z_4X_5$
g_3	$X_1X_3Z_4Z_5$
g_4	$Z_1X_2X_4Z_5$
$\bar{X}$	$XXXXX$
$\bar{Z}$	$ZZZZZ$

Tabela 2.3: Geradores do código de $[[5,1,3]]$ e os operadores lógicos $\bar{X}$ e $\bar{Z}$.

Dados os geradores, existe uma maneira alternativa de encontrar os *qubits lógicos* usando um processo iterativo. Tomamos como estado inicial $|\psi_1\rangle = |00000\rangle$, pois ele é estabilizado por $\bar{Z}$. O processo começa

com a aplicação $g_1|\psi_1\rangle$ que retorna $|10010\rangle$. O próximo estado será

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}\big(|00000\rangle + |10010\rangle\big).$$

Note que $|\psi_2\rangle$ é estabilizado por g_1. Selecionamos o segundo gerador, e repetimos o processo. A aplicação $g_2|\psi_2\rangle$ retorna $(|01001\rangle - |11011\rangle)/\sqrt{2}$. O próximo estado será

$$|\psi_2\rangle = \frac{1}{2}\big(|00000\rangle + |10010\rangle + |01001\rangle - |11011\rangle\big).$$

Continuamos o processo até encontrarmos um estado que será estabilizado por todos os geradores. O resultado final será

$$\begin{aligned}|0_L\rangle = \frac{1}{4}\big(&|00000\rangle + |10010\rangle + |01001\rangle - |11011\rangle + \\ &|10100\rangle - |00110\rangle - |11101\rangle - |01111\rangle + \\ &|01010\rangle - |11000\rangle - |00011\rangle - |10001\rangle - \\ &|11110\rangle - |01100\rangle - |10111\rangle + |00101\rangle\big).\end{aligned} \tag{2.5.12}$$

Agora aplicamos o operador $\bar{X}$ em $|0_L\rangle$

$$\begin{aligned}|1_L\rangle = \frac{1}{4}\big(&|11111\rangle + |01101\rangle + |10110\rangle - |00100\rangle + \\ &|01011\rangle - |11001\rangle - |00010\rangle - |10000\rangle + \\ &|10101\rangle - |00111\rangle - |11100\rangle - |01110\rangle - \\ &|00001\rangle - |10011\rangle - |01000\rangle + |11010\rangle\big).\end{aligned} \tag{2.5.13}$$

Note que $|0_L\rangle$ é estabilizado por $S_0 = \langle g_1, \cdots, g_4, \bar{Z}\rangle$, pois todos os *kets* de $|0_L\rangle$ têm um número par de 1s e $|1_L\rangle$ é estabilizado por $S_1 = \langle g_1, \cdots, g_4, -\bar{Z}\rangle$, pois todos os *kets* de $|1_L\rangle$ têm um número ímpar de 1s.

A Tabela 2.4 resume todos os possíveis erros de 1 qubit independentes (X e Z), mostra os resultados da análise de síndrome e os operadores de correção em cada caso. Na análise de síndrome do código de Shor, mostramos que quando um erro comuta com um gerador, o

Erro	Resultados das medidas	Correção
X_1	1, 1, 1,-1	X_1
X_2	-1, 1, 1, 1	X_2
X_3	-1,-1, 1, 1	X_3
X_4	1,-1,-1, 1	X_4
X_5	1, 1,-1,-1	X_5
Z_1	-1, 1,-1, 1	Z_1
Z_2	1,-1, 1,-1	Z_2
Z_3	1, 1,-1, 1	Z_3
Z_4	-1, 1, 1,-1	Z_4
Z_5	1,-1, 1, 1	Z_5

Tabela 2.4: Possíveis erros de 1 qubit no código $[[5, 1, 3]]$, análise de síndrome e os operadores de correção. Basta analisar os erros produzidos por X e Z nos diferentes qubits. Os resultados das medidas correspondem a medidas em cascata dos 4 geradores como observáveis.

resultado da medida com esse gerador é +1 e quando anti-comuta, o resultado é -1. Portanto, o resultado 1, 1, 1, -1 da síndrome do erro X_1 foi obtido da seguinte forma: X_1 comuta com g_1, g_2 e g_3 e anti-comuta com g_4. As outras síndromes foram obtidas de forma análoga. Os erros produzidos pela matriz de Pauli Y podem ser obtidos tomando o produto das linhas dos erros X por Z. Por exemplo, a linha correspondente a X_1 vezes a linha de Z_1 dá -1, 1, -1, -1 que corresponde ao erro produzido por Y_1. Note que há exatamente 16 possibilidades de resultados de medidas que correspondem exatamente aos erros produzidos pelas 16 matrizes de Pauli.

2.6 Código CSS no Formalismo Estabilizador

Vimos na Sec. 1.4 como definir os códigos CSS e usamos o código de Steane com um exemplo concreto. A análise de correção de erros e da codificação ficaram misteriosas naquela seção. O formalismo estabilizador facilita a descrição dessa classe de códigos, pois eles são aditivos. Vamos fazer uma análise mais detalhada dos códigos CSS no formalismo estabilizador.

Um código CSS requer dois códigos lineares clássicos $\mathcal{C}_1$ e $\mathcal{C}_2$ tais que $\mathcal{C}_2^\perp \subseteq \mathcal{C}_1$. Uma classe particular de códigos CSS de interesse prático é quando $\mathcal{C}_1 = \mathcal{C}_2 = \mathcal{C}$ e, portanto, $\mathcal{C}^\perp \subseteq \mathcal{C}$. Códigos lineares clássicos que satisfazem $\mathcal{C}^\perp \subseteq \mathcal{C}$ são chamados de *auto-duais*. Se $\mathcal{C}$ e $\mathcal{C}^\perp$ são do tipo (n, k, d) e $(n, n-k, d^\perp)$, respectivamente, o código quântico CSS será do tipo $[[n, 2k-n, d']]$. O valor de d' depende de vários fatores, mas se $n < 2k$, na maioria dos casos, temos $d' = d$. As palavras do código serão tratadas ora como *strings* de n bits, ora como vetores colunas de n componentes, ora como vetores linhas, dependendo do contexto. Vamos usar negrito para indicar explicitamente a notação vetorial.

O código clássico $\mathcal{C}$ é descrito por uma matriz verificadora de paridade H de dimensão $(n-k)\times n$ e uma matriz geradora G de dimensão $n \times k$, tal que $H\,G^T = 0$. As colunas de G geram as palavras do código e, se $\mathbf{v}$ é uma linha de H e $\mathbf{w}$ uma coluna de G, então $\mathbf{v}\cdot\mathbf{w} = 0$. A operação de produto interno é feita módulo 2. O código $\mathcal{C}^\perp$ é descrito pela matriz verificadora de paridade G^T e pela matriz geradora H^T. Pela hipótese de construção, as palavras geradas pelas colunas de H^T pertencem ao código $\mathcal{C}$. Por convenção, se H é uma matriz de paridade, a notação $\mathbf{v} \in H$ quer dizer que $\mathbf{v}$ é uma linha de H e, se G é uma matriz geradora, a notação $\mathbf{w} \in G$ quer dizer que $\mathbf{w}$ é uma coluna de G. Se $\mathbf{w} \in G$, então $\mathbf{w} \in \mathcal{C}$, porém a volta não precisa ser verdadeira.

O conjunto das palavras de $\mathcal{C}^\perp$ é um subgrupo do grupo aditivo $\mathcal{C}$ com a operação de soma bit-a-bit. Podemos então tomar uma transversal T de $\mathcal{C}^\perp$ em $\mathcal{C}$, cuja cardinalidade é o *índice* $|\mathcal{C} : \mathcal{C}^\perp| = |\mathcal{C}|/|\mathcal{C}^\perp|$, que é dado por 2^{2k-n}. As palavras do código CSS são as classes laterais de $\mathcal{C}^\perp$ em $\mathcal{C}$ indexadas pela transversal T, isto é, uma base para o código quântico CSS é $\{\left|\mathcal{C}^\perp + y\right\rangle \,|\, y \in T\}$, onde

$$\left|\mathcal{C}^\perp + y\right\rangle = \frac{1}{\sqrt{2^{n-k}}} \sum_{x\in\mathcal{C}^\perp} |x + y\rangle \tag{2.6.14}$$

e $x + y$ é a soma binária bit-a-bit das *strings* x e y, cada uma com n bits. Os vetores $\left|\mathcal{C}^\perp + y\right\rangle$ pertencem ao espaço de Hilbert de dimensão 2^n e geram um subespaço de dimensão 2^{2k-n}.

Para descrever o código CSS pelo formalismo estabilizador, temos que encontrar um subgrupo abeliano S do grupo de Pauli $\mathcal{G}_n$, tal que S tenha $2(n-k)$ geradores e $-I \notin S$. Os geradores de S podem ser obtidos através de H. A notação $X_{\mathbf{v}}$, onde $\mathbf{v}$ é um vetor binário de n componentes, descreve um operador de n qubits da seguinte forma: para cada 0 de $\mathbf{v}$, temos uma matriz identidade 2×2 e para cada 1, temos uma matriz X. Por exemplo,

$$X_{(0,1,1)} = I \otimes X \otimes X$$

que, nesse caso, atua em 3 qubits. Cada linha de H é um vetor binário de n componentes. Os $n-k$ primeiros geradores de S são $X_{\mathbf{v}}$, onde $\mathbf{v}$ são as linhas de H. Os $n-k$ geradores seguintes de S são $Z_{\mathbf{v}}$, onde $\mathbf{v}$ são as linhas de H. Portanto,

$$S = \langle X_{\mathbf{v}}, Z_{\mathbf{v}} \,|\, \mathbf{v} \in H \rangle. \tag{2.6.15}$$

Para mostrar que os geradores de S estabilizam as palavras do código CSS, temos que nos convencer primeiramente das seguintes igualdades:

$$X_{\mathbf{v}}|w\rangle = |w+v\rangle, \tag{2.6.16}$$
$$Z_{\mathbf{v}}|w\rangle = (-1)^{\mathbf{v}\cdot\mathbf{w}}|w\rangle, \tag{2.6.17}$$

onde $\mathbf{v} \cdot \mathbf{w}$ é o produto interno entre os vetores $\mathbf{v}$ e $\mathbf{w}$. A primeira igualdade é bastante simples. A atuação de $X_{\mathbf{v}}$ é equivalente a inverter os valores dos qubits onde as componentes de $\mathbf{v}$ são iguais a 1. Isso equivale a fazer a soma bit-a-bit com $\mathbf{v}$. A atuação de $Z_{\mathbf{v}}$ em $|w\rangle$ é equivalente a inverter o sinal do *ket* onde as componentes de $\mathbf{v}$ e $\mathbf{w}$ são iguais a 1 simultaneamente. Isso equivale multiplicar por $(-1)^{\mathbf{v}\cdot\mathbf{w}}$. Agora vamos verificar que $X_{\mathbf{v}}$ estabiliza as palavras do código CSS se $\mathbf{v} \in H$:

$$\begin{aligned} X_{\mathbf{v}}\big|\mathcal{C}^{\perp} + y\big\rangle &= \big|\mathcal{C}^{\perp} + y + v\big\rangle \\ &= \big|\mathcal{C}^{\perp} + y\big\rangle, \end{aligned}$$

pois, se $\mathbf{v} \in H$, então $\mathbf{v}^T \in H^T$. H^T é a matriz geradora de $\mathcal{C}^{\perp}$, portanto, $v \in \mathcal{C}^{\perp}$ e $\mathcal{C}^{\perp} + v = \mathcal{C}^{\perp}$. Analogamente, usando a Eq. (2.6.14),

obtemos

$$\begin{aligned} Z_{\mathbf{v}}\big|\mathcal{C}^{\perp}+y\big\rangle &= \frac{1}{\sqrt{2^{n-k}}}\sum_{x\in\mathcal{C}^{\perp}}(-1)^{\mathbf{v}\cdot(\mathbf{x}+\mathbf{y})}|x+y\rangle \\ &= \big|\mathcal{C}^{\perp}+y\big\rangle, \end{aligned}$$

pois como $\mathbf{v}\in H$ e $x+y\in\mathcal{C}$, segue que $\mathbf{v}\cdot(\mathbf{x}+\mathbf{y})=0 \mod 2$. Isso prova que os códigos CSS auto-duais são códigos estabilizadores.

O próximo passo é achar o grupo normalizador de S no grupo de Pauli $\mathcal{G}_n$, para os $2k-n$ operadores lógicos $\bar{X}$ e $\bar{Z}$ serem especificados. Os operadores lógicos $\bar{X}$ devem comutar entre si e com os operadores de S. O mesmo é válido para operadores lógicos $\bar{Z}$. No entanto, $\bar{X}$ deve anticomutar apenas com o operador $\bar{Z}$ correspondente. Para analisar as relações de comutação, temos que usar a seguinte igualdade:

$$X_{\mathbf{v}}Z_{\mathbf{w}}=(-1)^{\mathbf{v}\cdot\mathbf{w}}Z_{\mathbf{w}}X_{\mathbf{v}}, \tag{2.6.18}$$

que é válida porque temos um sinal negativo para cada componente igual a 1 simultânea em $\mathbf{v}$ e $\mathbf{w}$. Portanto, qualquer operador do tipo $X_{\mathbf{w}}$, onde $w\in\mathcal{C}\setminus\mathcal{C}^{\perp}$ ($w\in\mathcal{C}$ mas $w\notin\mathcal{C}^{\perp}$), comuta com todos operadores de S, pois $\mathbf{v}\cdot\mathbf{w}=0$ para todo $\mathbf{v}\in H$. Além disso, se $w\notin\mathcal{C}^{\perp}$, $X_{\mathbf{w}}$ não pertence a S. Portanto, $X_{\mathbf{w}}\in N(S)\setminus S$. O mesmo argumento vale para $Z_{\mathbf{w}}$, onde $w\in\mathcal{C}\setminus\mathcal{C}^{\perp}$. Portanto, $Z_{\mathbf{w}}\in N(S)\setminus S$. Os operadores lógicos $\bar{X}_{\mathbf{w}}$ e $\bar{Z}_{\mathbf{w}}$ serão esses operadores $X_{\mathbf{w}}$, $Z_{\mathbf{w}}$.

Exercício 2.3. *Mostre que $\big|\mathcal{C}^{\perp}\big\rangle$ é o único vetor, a menos de uma constante multiplicativa, estabilizado por*

$$\langle X_{\boldsymbol{v}}, Z_{\boldsymbol{v}}, \bar{Z}_{\boldsymbol{w}} \,|\, \boldsymbol{v}\in H, \boldsymbol{w}\in G\setminus\mathcal{C}^{\perp}\rangle.$$

Encontre os geradores para os subgrupos estabilizadores de $\big|\mathcal{C}^{\perp}+y\big\rangle$ onde $y\in G\setminus\mathcal{C}^{\perp}$.

As condições de correção de erros quânticos e sua relação com emaranhamento foram tratados nas Refs. [16, 2].

Sugestões para Leitura

Os códigos estabilizadores foram introduzidos por Daniel Gottesman na sua tese de doutoramento [10, 11] e pelos autores da Ref. [3]. A Ref. [20] tem excelente material sobre o assunto. O código quântico $[[5, 1, 3]]$ foi introduzido na Ref. [2].

Capítulo 3

Códigos Não-Aditivos

Os códigos estabilizadores não são os códigos ótimos no caso geral. Um código estabilizador $[[n, k, d]]$ é um subespaço vetorial de dimensão 2^k do espaço de Hilbert de dimensão 2^n. Para cada valor de n e d, existe um código estabilizador com um valor máximo de k. Por exemplo, para $n = 9$ e $d = 3$, o melhor código estabilizador é $[[9, 3, 3]]$, pois não existe nenhum código do tipo $[[9, k, 3]]$ com $k > 3$. Dependendo do valor de n, existem *códigos não-aditivos* $((n, K, d))$ com $K > 2^k$. A notação $((n, K, d))$ usada especialmente para códigos quânticos não-aditivos indica que o código tem dimensão K. Se $K > 2^k$, então o código não-aditivo codifica um espaço maior usando o mesmo número de qubits, tendo a mesma distância. Podemos dizer que ele é mais eficiente do que o melhor código estabilizador. Por exemplo, para $n = 9$ e $d = 3$, o melhor código quântico não-aditivo é $((9, 12, 3))$, cuja dimensão supera a dimensão do melhor código estabilizador, que é $2^3 = 8$.

Neste capítulo, vamos introduzir a noção de estado-grafo, que tem um papel importante tanto na área de códigos corretores de erros quânticos quanto na computação quântica como um todo. O estudo dos *estados-grafos* surgiu na área de *computação quântica direcionanda* (*one-way quantum computing*), mas imediatamente se espalhou para outros ramos da computação quântica. Esses estados são simples do ponto de vista algébrico, porém possuem diversas propriedades

interessantes.

O tema central do capítulo são os *códigos CWS* (*codeword stabilized codes*). Esse formalismo permite a construção sistemática de códigos quânticos não-aditivos a partir de *códigos clássicos*. A classe de códigos CWS inclui a classe dos códigos estabilizadores como um caso particular e é a técnica mais poderosa conhecida atualmente de geração de códigos não-aditivos. Quase todos os códigos quânticos não-aditivos conhecidos na literatura são do tipo CWS.

3.1 Estado-Grafos

Os circuitos da *porta Z-controlada*, ou *porta fase-controlada*, estão mostrados na Fig. 3.1. O *controle* e *alvo* podem ser trocados sem alterar o funcionamento da porta.

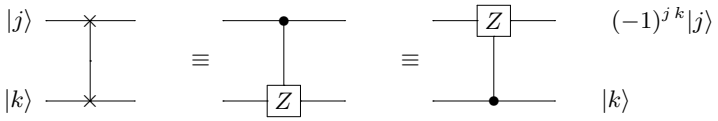

Figura 3.1: Representações do circuito da porta Z-controlada.

A representação algébrica dessa porta é

$$C_{1\,2}|j\rangle|k\rangle = (-1)^{j\,k}|j\rangle|k\rangle, \tag{3.1.1}$$

onde os sub-índices de C indicam em quais qubits a porta foi aplicada. Uma expressão equivalente para C é

$$C = \frac{1}{2}(I \otimes I + I \otimes Z + Z \otimes I - Z \otimes Z) \tag{3.1.2}$$

cuja expressão matricial é $C = \mathrm{diag}\{+1, +1, +1, -1\}$.

Se o computador quântico tem n qubits, podemos generalizar a descrição acima. A notação $Z_{\mathbf{v}}$, onde $\mathbf{v}$ é um vetor binário de n componentes, descreve um operador de n qubits da seguinte forma: para cada 0 de $\mathbf{v}$, temos uma matriz identidade 2×2 e para cada 1, temos uma matriz Z. Por exemplo,

$$Z_{(0,1,1)} = I \otimes Z \otimes Z$$

que atua em 3 qubits. Em particular, a notação Z_j, onde j é um número inteiro, deve ser entendida como $Z_{\mathbf{v}}$, onde $\mathbf{v}$ é o vetor cujas componentes são os dígitos binários de j. Por exemplo, $Z_7 = Z_{(1,1,1)}$. Nessa notação, a expressão algébrica da porta Z-controlada atuando nos qubits x_1 e x_2 é

$$C_{x_1x_2} = \frac{1}{2}\left(I + Z_{x_1} + Z_{x_2} - Z_{x_1}Z_{x_2}\right). \tag{3.1.3}$$

Seja $\Gamma(X, E)$ um *grafo não-direcionado* com o conjunto de *vértices* $X = \{x_1, \cdots, x_n\}$ ($|X| = n$) e conjunto das *arestas* E. Vamos associar cada vértice a um qubit na sequência usual. O estado-grafo Γ (*estado-grafo*) é definido da seguinte maneira: Primeiro inicializamos o computador quântico no *estado diagonal* da base computacional

$$|\mathrm{D}\rangle = |+\rangle^{\otimes n}. \tag{3.1.4}$$

Depois aplicamos uma porta Z-controlada para cada 2 qubits que sejam adjacentes. A ordem de aplicação não importa, pois essas portas comutam entre si. Algebricamente, definimos o operador U da seguinte forma:

$$U = \prod_{(x_j,x_k)\in E} C_{x_jx_k}, \tag{3.1.5}$$

onde $\prod$ indica o produto matricial das portas Z-controladas, uma porta para cada aresta (x_j, x_k) de E. O estado-grafo de Γ é

$$|\Gamma\rangle = U|\mathrm{D}\rangle. \tag{3.1.6}$$

É possível escrever uma expressão explícita do estado-grafo na base computacional. Suponha que a matriz de adjacência do grafo Γ seja M. Seja $\mathbf{v}$ um vetor binário onde a j-ésima componente é denotada por v_j. Então,

$$|\Gamma\rangle = \frac{1}{\sqrt{2^n}} \sum_{v_1=0}^{1} \cdots \sum_{v_n=0}^{1} (-1)^{\frac{1}{2}\mathbf{v}^T M \mathbf{v}} |v_1\rangle \cdots |v_n\rangle. \tag{3.1.7}$$

Essa expressão mostra que o estado-grafo na base computacional tem coeficientes $\pm 1/\sqrt{2^n}$.

Exercício 3.1. *Seja* $|\Gamma\rangle$ *o estado-grafo de* Γ. *Mostre que* $\langle\Gamma|Z_{\boldsymbol{v}}|\Gamma\rangle = 0$ *para qualquer vetor binário não-nulo* $\boldsymbol{v}$.

Exercício 3.2. *Seja* $|\Gamma\rangle$ *um estado-grafo. Vimos que* $\sqrt{2^n}|\Gamma\rangle$ *tem coeficientes* ± 1. *O inverso é verdadeiro, isto é, qualquer estado com coeficientes* ± 1 *é igual a* $\sqrt{2^n}|\Gamma\rangle$ *para algum grafo* Γ*? Dê um contra-exemplo.*

3.1.1 Subgrupo Estabilizador do Estado-Grafo

Na seção anterior descrevemos como obter o estado-grafo aplicando um operador unitário ao estado diagonal da base computacional. Sabemos que um estado quântico pode ser descrito a partir de um subgrupo estabilizador cujo número de geradores deve ser igual ao número de qubits. Existe uma maneira simples de achar o subgrupo estabilizador de um estado-grafo.

Seja $\Gamma(X, E)$ um grafo com o conjunto de vértices rotulados da seguinte forma: $X = \{1, \cdots, n\}$. Cada vértice vai determinar um gerador do subgrupo estabilizador. Tome o primeiro vértice. O gerador será da forma $X_1 Z_a Z_b \cdots$, onde $a, b, \cdots$ são os rótulos dos vértices que são adjacentes ao primeiro vértice. O mesmo procedimento deve ser feito para os outros vértices. Uma outra forma equivalente de descrever os geradores usa a *matriz de adjacência* do grafo. O j-ésimo gerador é dado por $X_j Z_{\mathbf{v}_j}$ onde $\mathbf{v}_j$ é a j-ésima linha da matriz de adjacência.

Por exemplo, considere o grafo desconexo de 2 vértices. O subgrupo estabilizador é $\{X_1, X_2\}$. O estado-grafo é $|\Gamma\rangle = |+\rangle|+\rangle$, pois este é o único vetor, módulo constante multiplicativa, que é *autovetor* simultâneo de X_1 e X_2 associado ao *autovalor* $+1$.

Considere o grafo da Fig. 3.2 como um exemplo mais elaborado. O subgrupo estabilizador é $S = \{g_1, g_2, g_3, g_4\}$, onde

$$\begin{aligned} g_1 &= X_1 Z_2 Z_4, \\ g_2 &= Z_1 X_2 Z_3 Z_4, \\ g_3 &= Z_2 X_3 Z_4, \\ g_4 &= Z_1 Z_2 Z_3 X_4. \end{aligned}$$

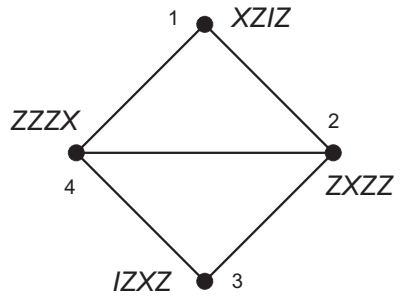

Figura 3.2: Exemplo de um grafo de 4 vértices com os respectivos geradores que estabilizam o estado-grafo.

A matriz de adjacência desse grafo é

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

Note que os índices dos operadores Z têm correspondência exata com as linhas da matriz de adjacência. Por exemplo, $g_1 = X_1 Z_{(0101)}$. Como $|0\rangle$ não é autovetor com autovalor -1 de nenhum dos geradores, o estado-grafo é obtido da seguinte forma:

$$\begin{aligned} |\Gamma\rangle &= \frac{1}{4}(I+g_1)(I+g_2)(I+g_3)(I+g_4)|0\rangle \\ &= \frac{1}{4}(+,+,+,-,+,-,-,-,+,-,+,+,-,-,+,-) \end{aligned} \tag{3.1.8}$$

onde os sinais indicam os coeficientes ± 1 da soma dos estados da base computacional $\{|0\rangle, \cdots, |15\rangle\}$.

Exercício 3.3. *Seja $|\Gamma_j\rangle$ o estado estabilizado pelo subgrupo gerado por $\{(-1)^{j_1} g_1, \cdots, (-1)^{j_n} g_n\}$, onde $j_1, \cdots, j_n$ é a decomposição binária de j. Mostre que o conjunto $\{|\Gamma_j\rangle,\ j = 0, \cdots, 2^n - 1\}$ é uma base ortonormal do espaço de Hilbert.*

3.2 Formulação Alternativa do Código [[5,1,3]]

Na Sec. 2.5, vimos o código quântico [[5,1,3]] através do formalismo estabilizador. Nessa seção, vamos mostrar como converter esse exemplo para um formalismo, chamado *Codeword Stabilizer* (CWS), que pode ser generalizado para códigos não-lineares. Na verdade, qualquer código estabilizador pode ser convertido para esse formalismo, porém nem todo código CWS é estabilizador.

O primeiro estado lógico do código [[5,1,3]] é

$$\begin{aligned}|0_L\rangle = \frac{1}{4}\big(&|00000\rangle + |10010\rangle + |01001\rangle - |11011\rangle + \\ &|10100\rangle - |00110\rangle - |11101\rangle - |01111\rangle + \\ &|01010\rangle - |11000\rangle - |00011\rangle - |10001\rangle - \\ &|11110\rangle - |01100\rangle - |10111\rangle + |00101\rangle\big).\end{aligned} \tag{3.2.9}$$

Ele é estabilizado pelo subgrupo

$$\begin{aligned}S \;=\; \langle &X_1Z_2Z_3X_4,\, X_2Z_3Z_4X_5,\, X_1X_3Z_4Z_5,\, Z_1X_2X_4Z_5, \\ &Z_1Z_2Z_3Z_4Z_5\rangle\,.\end{aligned} \tag{3.2.10}$$

Note que acrescentamos o operador lógico $\bar{Z}$ ao conjunto de forma que o número de geradores coincide com o número de qubits e, por convenção, essa é a maneira de escrever o subgrupo estabilizador do estado $|0_L\rangle$. Esse conjunto não é gerador de um estado-grafo, pois os operadores têm 2 Xs ou nenhum X. Note que não há outros elementos do subgrupo com um único X. A lista de todo os elementos não-triviais de S que não contém Y é

$$\begin{aligned}&-X_1X_2Z_4,\; -X_2X_3Z_5,\; -Z_1X_3X_4,\; -Z_2X_4X_5,\; -X_1Z_3X_5, \\ &X_1Z_2Z_3X_4,\; X_2Z_3Z_4X_5,\; X_1X_3Z_4Z_5,\; Z_1X_2X_4Z_5, \\ &Z_1Z_2X_3X_5,\; Z_1Z_2Z_3Z_4Z_5.\end{aligned} \tag{3.2.11}$$

Nenhum desses elementos pode ser gerador de um estado-grafo.

Nosso objetivo agora é achar um *operador unitário local*, isto é, produto tensorial de 5 operadores de 1 qubit, que transforme o estado

$|0_L\rangle$ em um estado-grafo. Se $|\Gamma\rangle = U|0_L\rangle$, onde $U = U_1U_2U_3U_4U_5$, então o subgrupo estabilizador de $|\Gamma\rangle$ é $S' = USU^\dagger$.

Queremos obter os elementos de S' por *conjugação* dos elementos de S de forma que cinco elementos tenham um único X. A ideia é converter simultaneamente X em Z e vice-versa, pois os cinco primeiros elementos da lista (3.2.11) conterão apenas um único X após a conjugação. O operador que leva por conjugação X em Z e vice-versa é o operador H, isto é, $HXH = Z$ e $HXH = Z$. Temos ainda que trocar os sinais desses geradores. Após conjugar com Hs, cada um dos cinco geradores conterão um único X. Para trocar o sinal, temos que conjugar com o operador Z, pois $ZXZ = -X$. Portanto, o operador U é dado por

$$U = (ZH)^{\otimes 5}$$

e os geradores do novo subgrupo estabilizador são

$$S' = \langle X_1Z_3Z_4,\ X_2Z_4Z_5,\ Z_1X_3Z_5,\ Z_1Z_2X_4,\ Z_2Z_3X_5\rangle\,. \qquad (3.2.12)$$

Esses geradores são independentes. Podemos agora obter a matriz de adjacência do grafo, que é dada por

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

O grafo é uma estrela conforme mostra a Fig. 3.3

A estrela é um ciclo com 5 vértices, cuja ordem dos rótulos em relação ao ciclo usual é 1,3,5,2,4. Se fizermos uma permutação de rótulos dada por $(2, 4, 5, 3)$, a estrela é convertida em um ciclo com a ordem usual 1,2,3,4,5. Essa permutação leva o subgrupo S' no novo subgrupo

$$S'' = \langle X_1Z_2Z_5,\ Z_1X_2Z_3,\ Z_2X_3Z_4,\ Z_3X_4Z_5,\ Z_1Z_4X_5\rangle\,. \qquad (3.2.13)$$

Permutação de rótulos é equivalente à permutação de qubits. A matriz

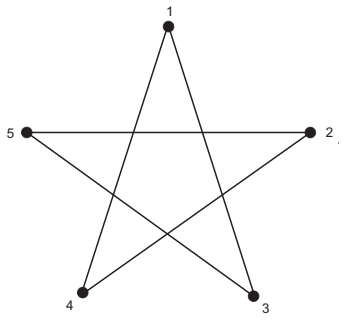

Figura 3.3: Grafo associado ao geradores de S' descrito na Eq. (3.2.12).

de adjacência associada a S'' é dada por

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

O grafo correspondente está mostrado na Fig. 3.4.

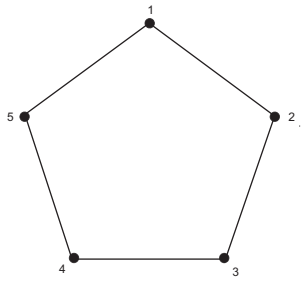

Figura 3.4: Grafo associado ao geradores de S'' descrito na Eq. (3.2.13).

O estado-grafo estabilizado por S'' é

$$|\omega_1\rangle = \frac{1}{4\sqrt{2}} \quad (\quad +,+,+,-,+,+,-,+,+,+,+,-,-,-,+,-, \\ +,-,+,+,+,-,-,-,-,+,-,-,+,-,-,- \quad). \tag{3.2.14}$$

Esse estado é o primeiro estado lógico do código CWS [[5,1,3]] no formato padrão.

Um *código CWS no formato padrão* requer a especificação de um grafo de n vértices cujo estado-grafo $|\omega_1\rangle$ é o primeiro estado lógico do código. Os outros estados lógicos são da forma

$$|\omega_j\rangle = Z_{\mathbf{v}_j}|\omega_1\rangle, \tag{3.2.15}$$

onde $Z_{\mathbf{v}_j}$ é chamado de um operador lógico. O conjunto de estados $\{|\omega_1\rangle, \cdots, |\omega_n\rangle\}$ forma uma base ortonormal para o código (ver Exercício 3.1). É importante notar que todos os outros estados lógicos são obtidos usando apenas operadores Z. Isso é um requisito essencial dos códigos CWS. A questão agora é saber como os vetores $\mathbf{v}_j$ são escolhidos? Vamos responder essa pergunta usando o exemplo do código [[5,1,3]], porém a estratégia é genérica.

O código [[5,1,3]] corrige erros em até um qubit e detecta erros em até 2 qubits. Já vimos anteriormente que basta analisar erros descritos em termos das matrizes X, Y e Z. No formalismo CWS, os erros X e Y podem ser convertidos em erros do tipo Z. Quando o grafo é uma estrela, um X é convertido em 2 Zs e um Y é convertido em 3 Zs.

Vamos supor que o erro seja X_2. O erro atua em um estado $|\psi\rangle$, que pode ser escrito em termos dos estado lógicos do código:

$$|\psi\rangle = \sum_{j=0}^{2^k-1} \psi_j|\omega_j\rangle, \tag{3.2.16}$$

onde ψ_j são as amplitudes e k é o número de qubits antes da codificação, que para o código $[[5,1,3]]$ é $k = 1$, ou seja, há dois estados lógicos. Basta analisar a atuação do erro na base. Se o erro X_2 atua no estado lógico $|\omega_j\rangle$, então

$$\begin{aligned} X_2|\omega_j\rangle &= X_2 Z_{\mathbf{v}_j}|\omega_1\rangle \\ &= X_2 Z_{\mathbf{v}_j} Z_1 X_2 Z_3|\omega_1\rangle \\ &= \pm Z_1 Z_3 Z_{\mathbf{v}_j}|\omega_1\rangle \\ &= \pm Z_1 Z_3|\omega_j\rangle, \end{aligned} \tag{3.2.17}$$

onde na primeira linha usamos a Eq. (3.2.15), na segunda linha usamos o fato de que $Z_1X_2Z_3$ é um gerador do subgrupo estabilizador que tem X atuando no segundo qubit, portanto, $|\omega_1\rangle = Z_1X_2Z_3|\omega_1\rangle$ e na terceira linha comutamos X_2 com $Z_{\mathbf{v}_j}$, que gera um sinal negativo se a segunda componente de $\mathbf{v}_j$ for 1. A matriz X desaparece e sobra um operador escrito em termos de 2 Zs. O operador X_2 é substituído por $\pm Z_1Z_3$ porque o vértice 2 está ligado aos vértices 1 e 3. Isso é uma característica geral. Se o erro for X_j, ele pode ser substituído por $\pm Z_{j-1}Z_{j+1}$, onde a operação de soma nos sub-índices é feita módulo 5.

Vamos agora considerar erros da forma Y_j. Usando a identidade $Y_j = iX_jZ_j$, podemos ver que um erro da forma Y_i pode ser convertido para $\pm Z_{j-1}Z_jZ_{j+1}$. Concluímos que, se o código CWS puder corrigir erros na forma $\pm Z_{\mathbf{w}_j}$, onde os possíveis valores de $\mathbf{w}_j$ dependem do grafo associado ao código, então ele corrige erros do tipo X e Y.

Como o código $[[5, 1, 3]]$ tem distância 3, ele deve corrigir um erro genérico em 1 qubit e deve detectar um erro genérico em até 2 qubits. Uma base dos possíveis erros em 1 qubit é Z_j, X_j e Y_j, $1 \leq j \leq 5$. Podemos expressar todos os erros apenas em termos de Z. Após a conversão, podemos usar uma notação binária onde 0 indica a identidade e 1 indica o operador Z, por exemplo, Z_1 é denotado por 10000. Todos os possíveis erros um 1 qubit são

10000	01000	00100	00010	00001
01001	10100	01010	00101	10010
11001	11100	01110	00111	10011.

Os erros em 2 qubits são obtidos através da soma bit-a-bit de dois erros de um qubit. Tomando todas as combinações possíveis obtemos os seguintes novos erros:

11000	01100	00110	00011	10001
11010	10110	10101	01101	01011
11110	11101	11011	10111	01111.

O objetivo agora é achar uma nova palavra cuja *soma bit-a-bit* com qualquer erro não dê a palavra 00000. Nesse caso a solução é única,

pois a única palavra restante é 11111. O operador lógico associado é $Z^{\otimes 5}$. Portanto, o código $[[5,1,3]]$ no formato CWS padrão tem como base os estados lógicos $|\omega_1\rangle$ e $|\omega_2\rangle$ onde $|\omega_1\rangle$ é dado por (3.4.24) e $|\omega_2\rangle = Z^{\otimes 5}|\omega_1\rangle$.

3.3 O Formalismo CWS

A construção de um código CWS $((n, K, d))$ no formato padrão se inicia com a escolha de um grafo Γ de n vértices. Os geradores do subgrupo estabilizador são

$$S = \langle X_j Z_{\mathbf{v}_j} \,|\, 1 \leq j \leq n \rangle,$$

onde $\mathbf{v}_j$ é a j-ésima linha da matriz de adjacência M do grafo Γ. Chamaremos o estado-grafo associado ao grafo Γ e estabilizado por S de $|\Gamma\rangle$.

As *palavras do código* são da forma

$$|\omega_k\rangle = \Omega_k |\Gamma\rangle, \tag{3.3.18}$$

onde Ω_k são operadores do grupo de Pauli $\mathcal{G}_n$ da seguinte forma

$$\Omega_k = Z_{\mathbf{w}_k}, \tag{3.3.19}$$

onde $\mathbf{w}_k$ são vetores binários de n componentes. O parâmetro k está no intervalo $1 \leq k \leq K$ e, em particular, para $k = 1$ impomos que $\Omega_1 = I$, $\mathbf{w}_1 = (0, \cdots, 0)$ e $|\omega_1\rangle = |\Gamma\rangle$.

Seja $\mathcal{E}$ o conjunto de erros que esse código detecta. Um operador genérico de $\mathcal{E}$ tem a forma $E = \pm X_{\mathbf{x}} Z_{\mathbf{z}}$, onde $\mathbf{x}$ e $\mathbf{z}$ são vetores binários de n componentes. Cada operador X presente no erro E pode ser eliminado usando a estratégia adotada na Eq. (3.2.17). Se houver um erro X_j atuando no j-ésimo qubit, ele pode ser cancelado as custas de introduzir novos operadores Z. De fato, suponha que o erro $E = \pm X_{\mathbf{x}} Z_{\mathbf{z}}$ é tal que a j-ésima componente de $\mathbf{x}$ é igual a 1. Se o erro E

está atuando em uma palavra do código, por exemplo, $|\omega_k\rangle$, então

$$\begin{aligned} E|\omega_k\rangle &= E\,\Omega_k|\Gamma\rangle \\ &= E\,\Omega_k X_j Z|\Gamma\rangle \\ &= \pm(E\,X_j Z_{\mathbf{v}_j})\,\Omega_k|\Gamma\rangle \\ &= \pm(E\,X_j Z_{\mathbf{v}_j})|\omega_k\rangle, \end{aligned}$$

onde usamos a Eq. (3.3.18) na primeira linha, o fato que $X_j Z_{\mathbf{v}_j}$ estabiliza $|\Gamma\rangle$ na segunda linha, o fato que dois operadores do grupo de Pauli comutam ou anti-comutam na terceira linha e novamente usamos a Eq. (3.3.18) na última linha. O erro E é equivalente a $\pm E\,X_j Z_{\mathbf{v}_j}$, que não contém X_j. Devemos realizar essa operação para cada componente igual a 1 em $\mathbf{x}$. No final, teremos um erro sem o operador X, isto é, o erro passará a ser $E' = \pm Z_{\mathbf{u}}$ onde

$$\mathbf{u} = \mathbf{z} + \sum_{j=1}^{n} x_j \mathbf{v}_j, \tag{3.3.20}$$

$\mathbf{v}_j$ é a j-ésima linha da matriz de adjacência M e x_j é a j-ésima componente de $\mathbf{x}$. A soma dos vetores é feita modulo 2.

O formalismo CWS visa expressar tanto as palavras do código como os erros em termos de *vetores binários*. Se um conjunto de erros quânticos $\mathcal{E}$ for dado, podemos converter esse conjunto de operadores em um conjunto de vetores binários usando a Eq. (3.3.20). A partir desse conjunto de erros clássicos temos que achar os vetores binários $\mathbf{w}_k$ da Eq. (3.3.19). Se os vetores $\mathbf{w}_k$ forem palavras de um código clássico que corrige o conjunto de *erros clássicos* associado ao conjunto de erros quânticos, o código CWS corrige os erros quânticos. Portanto, no formalismo CWS, o problema de encontrar um código quântico se reduz ao problema de encontrar *códigos clássicos*.

Para mostrar a relação entre a correção clássica e a quântica, vamos supor que o código CWS obedece às *condições quânticas de detecção de erros*, isto é,

$$\langle\omega_i|E|\omega_j\rangle = c_E\delta_{i,j}, \tag{3.3.21}$$

para todo $E \in \mathcal{E}$ onde c_E depende apenas de E. Essas condições querem dizer que o erro E será detectado se, e somente se, a Eq. (3.3.21)

for satisfeita para todo i, j. Usando a Eq. (3.3.18) obtemos

$$\langle\Gamma|\Omega_i^\dagger E\,\Omega_j|\Gamma\rangle = c_E\delta_{i,j}. \tag{3.3.22}$$

Se $i \neq j$, as condições se resumem a $\langle\Gamma|\Omega_i^\dagger E\,\Omega_j|\Gamma\rangle = 0$, que é equivalente a

$$\Omega_i^\dagger E\,\Omega_j \notin \pm S, \tag{3.3.23}$$

para todo i, j, pois se $\Omega_i^\dagger E\,\Omega_j \in \pm S$, $\langle\Gamma|\Omega_i^\dagger E\,\Omega_j|\Gamma\rangle = \pm 1$ e se $\Omega_i^\dagger E\,\Omega_j \notin \pm S$, $\Omega_i^\dagger E\,\Omega_j$ não estabiliza $|\Gamma\rangle$ e $-|\Gamma\rangle$. Essas condições também são suficientes quando $i \neq j$, pois tanto Ω_j como E são operadores do grupo de Pauli, eles comutam ou anti-comutam.

A partir desses resultados, vamos mostrar agora que o código CWS detecta erros no conjunto $\mathcal{E}$ se, e somente se, existe um código clássico associado que detecta os erros clássicos associados, além disso, para cada E, temos que o erro clássico associado a E é diferente de zero ou $Z_{\mathbf{w}_k}E = EZ_{\mathbf{w}_k}$, $\forall k$. De fato, quando $i \neq j$, $\Omega_i^\dagger E\,\Omega_j \notin \pm S$ é verdadeiro quando $Z_{\mathbf{w}_i}^\dagger E\mathbb{Z}_{\mathbf{w}_j} \notin \pm S$, o que é equivalente a $Z_{\mathbf{w}_i}^\dagger Z_{\mathbf{e}}\mathbb{Z}_{\mathbf{w}_j} \notin \pm S$, onde $\mathbf{e}$ é o erro clássico associado a E. Na forma padrão, o único elemento de S sem X é a identidade, de forma que isto é satisfeito exatamente quando $\mathbf{w}_i \oplus \mathbf{e} \neq \mathbf{w}_j{}'$, que são as condições de correções clássicas. Para completar a análise, devemos considerar o caso $i = j$. Isso é deixado com exercício.

O exemplo a seguir detalha todo o processo de obtenção de um código CWS. O exemplo também mostra um algoritmo para o cálculo do código CWS ótimo, quando fixamos os parâmetros do código.

3.4 Exemplo

Nesta seção, vamos ver um exemplo de código CWS não-aditivo. O primeiro código quântico não-aditivo descrito na literatura foi o código $((5, 6, 2))$, cuja notação indica que o código tem dimensão 6, usa 5 qubits e tem distância 2. Esse código é superior ao melhor código estabilizador de 5 qubits e distância 2, que é o código $[[5, 2, 2]]$.

O código $((5, 6, 2))$ é um código CWS e, portanto, pode ser obtido através do formalismo CWS. O ponto de partida para gerar um

código CWS, após fixar n e d, é a escolha do grafo. O grafo vai determinar todos os resultados posteriores e a complexidade das tarefas. No exemplo, vamos usar como grafo um ciclo de 5 vértices, cuja matriz de adjacência é

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

Portanto, os geradores são

$$S = \langle XZIIZ, ZXZII, IZXZI, IIZXZ, ZIIZX \rangle.$$

O estado-grafo é dado por

$$|\Gamma\rangle = \frac{1}{4\sqrt{2}} \quad (\quad +,+,+,-,+,+,-,+,+,+,+,-,-,-,+,-, \\ +,-,+,+,+,-,-,-,-,+,-,-,+,-,-,- \quad). \tag{3.4.24}$$

A próxima etapa é gerar o código clássico associado. Note que as palavras do códigos são da forma $|\omega_k\rangle = Z_{\mathbf{w}_k}|\Gamma\rangle$, $1 \leq k \leq 6$, sendo $\mathbf{w}_1 = (0,0,0,0,0)$ obrigatoriamente. Queremos encontrar os $\mathbf{w}_k$ restantes no universo de busca, que é $\mathbb{Z}_2^5$. Para isso, vamos analisar que erros esse código deve detectar obrigatoriamente. Como $d = 2$, os erros devem ter peso de Hamming 1. Portanto,

$$\mathcal{E} = \{Z_j, X_j, Y_j,\ 1 \leq j \leq 5\}.$$

Esses erros devem ser convertidos para erros do tipo clássico, ou seja, do tipo $\pm Z_{\mathbf{u}}$, através da Eq. (3.3.20). O conjunto dos vetores $\mathbf{u}$ formam o conjunto de erros do código clássico associado. Usando a matriz

de adjacência do ciclo, obtemos

$$\begin{aligned}\mathcal{E} = \{&(1,0,0,0,0),(0,1,0,0,0),(0,0,1,0,0),\\ &(0,0,0,1,0),(0,0,0,0,1),(0,1,0,0,1),\\ &(1,0,1,0,0),(0,1,0,1,0),(0,0,1,0,1),\\ &(1,0,0,1,0),(1,1,0,0,1),(1,1,1,0,0),\\ &(0,1,1,1,0),(0,0,1,1,1),(1,0,0,1,1)\}.\end{aligned}$$

Objetivo agora é obter um código clássico com 6 palavras, que corrige o conjunto de erros acima. Esse código será necessariamente não-linear, ou seja, as palavras do código não formam um grupo. Códigos menores vão existir, porém eles não são interessantes, pois são *subcódigos* e, portanto, não-ótimos. Existe um procedimento sistemático para se obter o código ótimo. Devemos montar um grafo, chamado de *supergrafo*, cujos *rótulos dos vértices* são os vetores binários de $\mathbb{Z}_2^5 \setminus \mathcal{E}$. Esses rótulos, tanto no formato de *string* binárias como decimais, são

$$\begin{array}{llll} 00000 = 0 & 00011 = 3 & 00110 = 6 & 01011 = 11 \\ 01100 = 12 & 01101 = 13 & 01111 = 15 & 10001 = 17 \\ 10101 = 21 & 10110 = 22 & 10111 = 23 & 11000 = 24 \\ 11010 = 26 & 11011 = 27 & 11101 = 29 & 11110 = 30 \\ 11111 = 31. & & & \end{array}$$

O supergrafo tem $32 - 15 = 17$ vértices. Dois vértices $\mathbf{v}_1$,$\mathbf{v}_2$ são adjacentes se, e somente se, $\mathbf{v}_1 \oplus \mathbf{v}_2 \notin \mathcal{E}$. Isso serve para garantir que o código vai satisfazer as condições de correção de códigos clássicos, pois se $\mathbf{u}$ é um erro, então a soma $\mathbf{v} \oplus \mathbf{u}$ de uma palavra $\mathbf{v}$ do código com um erro detectável qualquer não pode ser uma palavra do código, ou seja, $\mathbf{v}_1 \oplus \mathbf{u} \neq \mathbf{v}_2$, para quaisquer palavras $\mathbf{v}_1$,$\mathbf{v}_2$ do código. Isso é equivalente a $\mathbf{v}_1 \oplus \mathbf{v}_2 \notin \mathcal{E}$. O supergrafo gerado pelos erros acima está mostrado na Fig. 3.5.

O supergrafo pode ser implementado computacionalmente e existem diversos programas prontos para encontrar uma *clique maximal*, isto é, um subgrafo completo maximal. A clique maximal não é necessariamente única. Um exemplo de clique maximal do supergrafo acima é

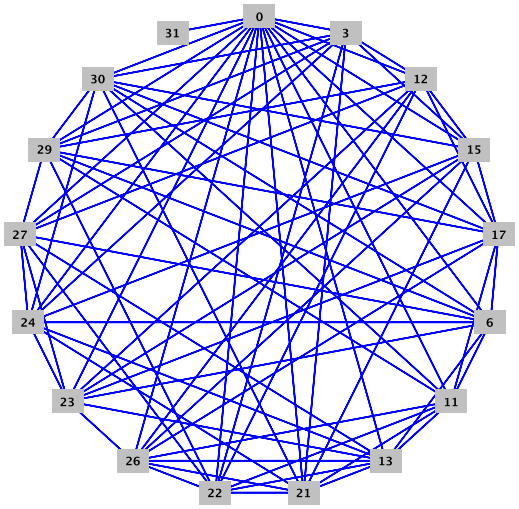

Figura 3.5: Supergrafo do código $((5, 6, 2))$.

$$\begin{array}{lll} 00000 = 0 & 01011 = 11, & 01101 = 13 \\ 10101 = 21 & 10110 = 22 & 11010 = 26 \end{array}$$

Portanto, o código $((5, 6, 2))$ é gerado pelos vetores

$$\begin{aligned} |\omega_1\rangle &= |\Gamma\rangle, \\ |\omega_2\rangle &= Z_2Z_4Z_5|\Gamma\rangle, \\ |\omega_3\rangle &= Z_2Z_3Z_5|\Gamma\rangle, \\ |\omega_4\rangle &= Z_1Z_3Z_5|\Gamma\rangle, \\ |\omega_5\rangle &= Z_1Z_3Z_4|\Gamma\rangle, \\ |\omega_6\rangle &= Z_1Z_2Z_4|\Gamma\rangle. \end{aligned}$$

Exercício 3.4. *Mostre que os vértices 0, 3, 12, 22, 27, 29 também formam um clique maximal. Quais são as palavras do código nesse caso?*

Exercício 3.5. *Use o formalismo CWS para mostrar que todos os códigos ótimos com $n = 4$ e $d = 2$ são estabilizadores.*

Exercício 3.6. *Use o formalismo CWS para mostrar que todos os códigos ótimos com* $n = 5$ *e* $d = 3$ *são estabilizadores.*

Sugestões para Leitura

O primeiro código não-aditivo com distância $d \geq 3$ apareceu na Ref. [28], cujos parâmetros são $((9, 12, 3))$. Esse código supera o melhor código estabilizador de parâmetros equivalentes, que é o códigos $[[9, 3, 3]]$ descrito na Ref. [5]. Esse resultado foi generalizado gerando o modelo *Codeword Stabilized Codes* (CWS) binário nas Refs. [29, 8, 7] e não-binário nas Refs. [14, 6]. A concatenação de códigos CWS foi discutida na Ref. [12, 1]. A decodificação de códigos CWS é analisada nas Refs. [18, 19].

Apêndice A

Teoria de Grupos

O objetivo deste apêndice é compilar as definições, notações e fatos da Teoria de Grupos que são importantes neste trabalho. Como a teoria de grupos não é matéria obrigatória nas diversas áreas das ciências exatas, acrescentamos diversos exercícios para tornar o material mais didático. Vamos nos concentrar em grupos finitos. Faremos muitas afirmações sem apresentar as demonstrações. Algumas demonstrações são pedidas nos exercícios. As outras podem ser encontradas nas referências caso o leitor tenha interesse.

A.1 Definições Básicas

Um *grupo* G é um conjunto não-vazio com uma operação binária de multiplicação "$\cdot$" que satisfaz às seguintes propriedades:

- $g_1 \cdot g_2 \in G$, $\forall g_1, g_2 \in G$ (fechamento);
- $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$, $\forall g_1, g_2, g_3 \in G$ (associatividade);
- $\exists\, e \in G$ tal que $\forall g \in G$, $e \cdot g = g \cdot e = g$ (existência da identidade);
- $\forall g \in G$ existe $g^{-1} \in G$ tal que $g^{-1} \cdot g = g \cdot g^{-1} = e$ (elemento inverso).

Dependendo do contexto, omitiremos a notação "$\cdot$" para a operação binária, por exemplo, $g_1 g_2$ no lugar de $g_1 \cdot g_2$. Vamos usar outros

símbolos como, por exemplo "+" quando a ordem dos fatores não altera o resultado. Nesse caso, o grupo é dito *abeliano* ou *comutativo*.

Alguns fatos decorrem imediatamente dos postulados acima:

- O elemento identidade e é único;
- Para cada $g \in G$, existe um único elemento inverso $g^{-1} \in G$;
- Se $g \in G$, $(g^{-1})^{-1} = g$;
- $(g_1 g_2)^{-1} = g_2^{-1} g_1^{-1}$, $\forall g_1, g_2 \in G$.

Neste livro vamos lidar com grupos finitos, portanto alguns fatos não poderão ser aplicados no caso geral. A *ordem de um grupo* G, denotada por $|G|$, é o número de elementos de G. A *ordem de um elemento* $g \in G$, denotada por $|g|$, é o menor inteiro positivo r tal que $g^r = e$.

Um *subgrupo* H de G é um subconjunto de G que forma um grupo sob a mesma operação binária de G. Usamos a notação $H \leq G$ para indicar que H é subgrupo de G. Todo grupo tem dois subgrupos triviais, a saber, o próprio grupo e o conjunto $\{e\}$. Para mostrar que um subconjunto H de G finito é um subgrupo, basta mostrar que a operação binária é fechada em H. As outras propriedades serão satisfeitas automaticamente. Com essas definições básicas é possível provar o *teorema de Lagrange*: *Se H é um subgrupo de G, $|H|$ divide*[1] $|G|$.

Se H é um subgrupo de G, uma *classe lateral à esquerda* de H em G com representante g é o conjunto $gH = \{gh | h \in H\}$. A *classe lateral à direita* é definida de forma similar. Quando o grupo G é comutativo, as classes laterais à esquerda e à direita com representante g são iguais. Nesse caso, usamos apenas a denominação *classe lateral*. Qualquer elemento de uma classe lateral pode ser usado como representante. O número de classes laterais é chamado de *índice* de H em G denotado

[1] Em teoria dos números diz-se que um número inteiro não nulo a divide b se existe um inteiro c tal que $b = a.c$. Se a divide b, b é dito um múltiplo de a e a é chamado um divisor de b. Se a divide b usamos o símbolo $a|b$.

por $|G : H|$. As classes laterais de H em G têm a mesma cardinalidade e um corolário do teorema de Lagrange é $|G|/|H| = |G : H|$. Um *conjunto transversal à esquerda* de H em G é um subconjunto de G de cardinalidade $|G : H|$ onde cada elemento pertence a uma classe lateral à esquerda distinta de H em G. O *conjunto transversal à direita* é definido de forma análoga. Se $\{g_1, \cdots, g_k\}$ é uma transversal à esquerda de H em G, então $G = g_1H \cup \cdots \cup g_kH$ e $g_iH \cap g_jH = \varnothing$ $\forall i \neq j$.

Exercício A.1. *1. Mostre que o conjunto*

$$G = \{\pm I, \pm Z, \pm X, \pm iY\},$$

onde

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

é um grupo com a operação de multiplicação usual de matrizes. Use o fato de que o conjunto das matrizes 2×2 inversíveis é um grupo.

2. G é comutativo?

3. Qual é a ordem de cada elemento de G?

4. Exiba um subgrupo H de ordem 4. Existe algum outro?

5. Exiba as classes laterais à esquerda e à direita de H em G. São iguais?

6. Qual é o índice de H em G.

7. Exiba uma transversal de H em G.

Exercício A.2. *Se G_1 e G_2 são grupos, mostre que o produto cartesiano $G_1 \times G_2$ com a operação binária de multiplicação definida por $(g_1, g_2)(h_1, h_2) = (g_1h_1, g_2h_2)$, é um grupo. $G_1 \times G_2$ é chamado produto direto de G_1 por G_2.*

A.2 Grupos Cíclicos e Geradores

Um grupo G é dito *cíclico* se existe um elemento $g \in G$ tal que $G = \{g^i | i \in \mathbb{Z}\}$. O elemento g é chamado de *gerador* de G. A ordem de g é igual a $|G|$. Sempre que mencionarmos geradores, vamos usar a notação $\langle g \rangle$ no lugar de $\{g^i | i \in \mathbb{Z}\}$. Se $g \neq e$, a ordem de G é maior ou igual a 2.

No caso geral, um subconjunto $\{g_1, \cdots, g_k\}$ de G é chamado de *conjunto gerador* do grupo G, se todo elemento de G pode ser escrito como um produto de elementos de $\{g_1, \cdots, g_k\}$, sendo permitido repetições. De forma compacta, escrevemos $G = \langle g_1, \cdots, g_k \rangle$. O conjunto gerador é análogo ao conceito de base de *espaços vetoriais*. A grande vantagem de expressar um grupo através de um conjunto gerador reside na economia de recursos computacionais para manipular e realizar cálculos com os elementos do grupo. Um grupo de ordem $|G|$ pode ser gerado por um conjunto gerador cuja cardinalidade é menor ou igual a $\lceil \log_2 |G| \rceil$. Um conjunto gerador $\{g_1, \cdots, g_k\}$ é *minimal*, se, ao retirarmos um elemento, por exemplo g_k, o conjunto restante $\{g_1, \cdots, g_{k-1}\}$ não gera do grupo G. Nesse caso, $\{g_1, \cdots, g_{k-1}\}$ gera um subgrupo de G cuja cardinalidade é menor ou igual a $|G|/2$. O elemento identidade e não pode pertencer ao conjunto gerador minimal. Vamos usar a denominação *conjunto gerador* como sinônimo de *conjunto gerador minimal* a menos que explicitamente dito ao contrário.

Exercício A.3. *Mostre que se G é um grupo finito, então $g^{|G|} = e$ para todo $g \in G$.*

Exercício A.4. *Mostre que se G é um grupo finito, então a ordem de qualquer elemento de G divide $|G|$.*

Exercício A.5. *Mostre que qualquer grupo, cuja ordem é um número primo, é cíclico.*

Exercício A.6.

1. *Suponha que G é um grupo gerado por 2 geradores. Mostre que $|G| \geq 4$.*
2. *Mostre que um grupo de ordem $|G|$ pode ser gerado por um conjunto gerador cuja cardinalidade é menor ou igual a $\lceil \log_2 |G| \rceil$.*

Exercício A.7. *Seja $\langle g_1, \cdots, g_k \rangle$ um conjunto gerador de G. Mostre que G é comutativo se, e somente se $g_i g_j = g_j g_i$ para todo i, j.*

A.3 Homomorfismos

Sejam G e G' grupos, cada um com sua respectiva operação binária. A função $\phi : G \to G'$ é um *homomorfismo* se $\phi(g_1 g_2) = \phi(g_1)\phi(g_2)$. A multiplicação $g_1 g_2$ é feita com a operação binária de G, enquanto que a multiplicação $\phi(g_1)\phi(g_2)$ é feita com a operação binária de G'. A função ϕ não precisa ser sobrejetiva.

Se o homomorfismo ϕ for sobrejetivo e injetivo, então ϕ é chamado de *isomorfismo*. Se existir um isomorfismo $\phi : G \to G'$, G e G' são grupos *isomorfos* e usamos a notação $G \simeq G'$. Se $G' = G$, o isomorfismo é chamado de *automorfismo*.

Por exemplo, seja G um grupo e selecione $g \in G$. Defina a função $\phi : G \to G$ por $\phi(h) = g^{-1}hg$ para todo $h \in G$. A função ϕ é um automorfismo. Para verificar esse fato, temos que mostrar que ϕ é um homomorfismo sobrejetivo um-a-um. Ele é um homomorfismo porque

$$\phi(h_1 h_2) = g^{-1}h_1 h_2 g = g^{-1}h_1 g \cdot g^{-1}h_2 g = \phi(h_1)\phi(h_2).$$

Ele é sobrejetivo porque

$$h = g^{-1}(ghg^{-1})g = \phi(ghg^{-1}),$$

para qualquer $h \in G$. Um automorfismo $\phi : G \to G$ dado por $\phi(h) = g^{-1}hg$ é chamado de *automorfismo interno* de G induzido por g.

Os seguinte fatos são importantes: se $\phi : G \to G'$ é um homomorfismo, então $\phi(e) = e'$, onde e' é o elemento identidade de G', e $\phi(g^{-1}) = \phi(g)^{-1}$, para todo $g \in G$. O *núcleo* de ϕ, denotado por $\mathrm{Ker}(\phi)$, é o conjunto $\{g \in G | \phi(g) = e'\}$. O núcleo de ϕ é um subgrupo de G e $h^{-1}\mathrm{Ker}(\phi)h \subseteq \mathrm{Ker}(\phi)$ para todo $h \in G$.

Exercício A.8. *Mostre que o núcleo de um homomorfismo $\phi : G \to G'$ é um subgrupo de G e $h^{-1}Ker(\phi)h \subseteq Ker(\phi)$ para todo $h \in G$.*

Exercício A.9. *Para quais elementos $g \in G$ a função $\psi : G \to G$, dada por $\psi(h) = hg$ para todo $h \in G$, é um automorfismo?*

Exercício A.10. *Seja G um grupo. Cada elemento $g \in G$ induz um automorfismo interno de G. Mostre que o conjunto de todos os automorfismos internos, denotado por Inn(G), forma um grupo com a operação binária de composição.*

Exercício A.11. *1. Seja G um grupo. Mostre que o conjunto de todos os automorfismos de G, denotado por Aut(G), forma um grupo com a operação binária de composição.*

2. Mostre que $Inn(G) \lhd Aut(G)$.

A.4 Grupos de Ordem Pequena

A quantidade de grupos de ordem menor ou igual a oito é surpreendentemente baixa módulo isomorfismos. Em função disso, vamos listá-los e nomeá-los.

Ordem 1

Só existe um grupo: $\{e\}$.

Ordem 2, 3, 5 e 7

Como a ordem é um número primo, só existe um grupo de cada ordem, a saber, $\mathbb{Z}_2$, $\mathbb{Z}_3$, $\mathbb{Z}_5$ e $\mathbb{Z}_7$, respectivamente.

Ordem 6

Só existem 2 grupos não-isomorfos: $\mathbb{Z}_6$ e S_3 (grupo simétrico[2] de 3 pontos).

[2]Uma permutação de um conjunto A é uma função bijetiva de A em A. O conjunto formado por todas as permutações de A forma um grupo com a operação de composição de funções. Seja $A = \{1, 2, \ldots, n\}$, ao conjunto das permutações de A com a composição dá-se o nome de grupo Simétrico de grau n, e é denotado por S_n.

Ordem 8

Só existem 5 grupos não-isomorfos, 3 abelianos e 2 não-abelianos: $\mathbb{Z}_8$, $\mathbb{Z}_4 \times \mathbb{Z}_2$, $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$, D_4 (grupo diedral das simetrias de um quadrado) e Q_3 (grupo dos quatérnios).

Exercício A.12. *O grupo analisado no Exercício A.1 é isomorfo a que grupo listado acima?*

A.5 Subgrupos Normais

Um subgrupo N de G é dito *normal*, se $g^{-1}Ng \subseteq N$ para todo $g \in G$. A notação $N \lhd G$ indica que N é um subgrupo normal de G. Os exemplos triviais de subgrupos normais de G são o conjunto $\{e\}$ e o próprio grupo G. Um exemplo menos trivial é o núcleo de um homomorfismo $\phi : G \to G'$. Temos o seguinte fato: $\mathrm{Ker}(\phi) \lhd G$.

O núcleo de um homomorfismo é um subgrupo normal de G e dado um subgrupo normal de G podemos sempre definir um homomorfismo que tem esse subgrupo como núcleo do homomorfismo. Portanto, existe uma equivalência entre as noções de homomorfismo e subgrupo normal.

Na definição de subgrupo normal exigimos que $g^{-1}Ng \subseteq N$ em vez de $g^{-1}Ng = N$. No entanto, essa última igualdade é sempre válida, porém exige mais trabalho para demonstrar. Uma consequência imediata dessa igualdade é que as classes laterais à esquerda coincidem com as classes laterais à direita quando N é um subgrupo normal de G. Esse fato pode ser usado como uma definição alternativa de subgrupo normal. Desse ponto de vista, fica simples de mostrar que um subgrupo de G cuja ordem é $|G|/2$ é normal em G. Por outro lado, usando a definição usual, é trivial mostrar que todo subgrupo de um grupo comutativo é normal.

A.6 Grupos Quocientes

Seja N um subgrupo normal do grupo G. Seja $\{g_1, \cdots, g_k\}$ uma transversal de H em G, onde $k = |G|/|N|$ é o índice de H em G. Vamos denotar esse conjunto transversal por G/N e definir uma operação binária de multiplicação da seguinte forma: sabemos que $g_i g_j$ pertence a G, portanto, $g_i g_j$ pertence a uma das classes laterais de N em G. Suponha que o representante dessa classe lateral seja g_m, então $g_i g_j = g_m$. O conjunto G/N com essa operação binária forma um grupo, chamado de *grupo quociente*. Observe que o grupo quociente não herda a operação do grupo G, portanto para mostrar que G/N é um grupo temos que verificar o fechamento, a associatividade, a existência de elemento identidade e a existência de elemento inverso. Além disso, é importante que a construção do grupo G/N não dependa da escolha dos representantes na transversal. Todos esses requisitos são satisfeitos como a definição acima.

Uma definição alternativa do grupo G/N é $\{Ng | g \in G\}$. Os elementos do conjunto $\{Ng | g \in G\}$ são as classes laterais de N em G, pois $Nh_1 = Nh_2$ se, e somente se h_1 e h_2 pertencem a mesma classe lateral. O produto de duas classes laterais é definido como $(Nh_1)(Nh_2) = N(h_1 h_2)$. Em particular, $N \in \{Ng | g \in G\}$ e N é o elemento identidade de G/N. Essa definição alternativa é útil para provar que G/N é um grupo.

O grupo G/N permite a construção do seguinte homomorfismo: seja $\phi : G \to G/N$ tal que $\phi(g) = Ng$. O núcleo de ϕ é N.

A.7 Centro, Centralizador e Normalizador

O *centro* de um grupo G, denotado por $Z(G)$ é o conjunto $\{h \in G \,|\, g^{-1}hg = h, \forall g \in G\}$. Os elemento do centro comutam com todos elementos de G. O centro é um subgrupo abeliano normal de G. Se G for abeliano, o centro é o próprio G. Por outro lado, existem grupos cujo centro é o grupo trivial $\{e\}$. Vale o seguinte fato: $\mathrm{Inn}(G) \simeq G/Z(G)$, ou seja, o grupo do automorfismos internos de G é isomorfo ao grupo quociente $G/Z(G)$.

Seja G um grupo e H um subconjunto de G. O *centralizador* de H em G, denotado por $C_G(H)$ é o conjunto $\{g \in G \,|\, g^{-1}hg = h,\ \forall h \in H\}$. $C_G(H)$ é um subgrupo de G. O centro de G é $C_G(G)$.

Seja G um grupo e H um subconjunto de G. O *normalizador* de H em G, denotado por $N_G(H)$ é o conjunto $\{g \in G \,|\, g^{-1}Hg = H\}$. $N_G(H)$ é um subgrupo de G. Vale o seguinte fato: $C_G(H) \leq N_G(H)$.

Exercício A.13. *1. Encontre o centro do grupo $\langle Z, X\rangle$.*

2. Encontre o centro do grupo $\langle Z, X, iI\rangle$. O centro é isomorfo a que grupo da Sec. A.4?

A.8 Grupo de Pauli

O *grupo de Pauli* $\mathcal{G}_n$ é um conjunto de 4^{n+1} matrizes de dimensão 2^n, onde n é um número inteiro. Quando $n = 1$, o grupo de Pauli é

$$\mathcal{G}_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{A.8.1}$$

É fácil de verificar que este conjunto satisfaz as propriedades de fechamento, existência de elemento neutro ($I_{2\times 2}$), associatividade e existência de elemento inverso (próprio elemento ou o negativo do elemento) com relação ao produto usual de matrizes. O grupo $\mathcal{G}_1$ é não comutativo e $\mathcal{G}_1 = \langle X, Z, iI\rangle$.

Quando $n = 2$, o grupo $\mathcal{G}_2$ é obtido tomando o produto tensorial dos elementos de $\mathcal{G}_1$, isto é

$$\mathcal{G}_2 = \{\pm I\otimes I, \pm iI\otimes I, \pm I\otimes X, \pm iI\otimes X, \pm I\otimes Y, \cdots, \pm iZ\otimes Z\}. \tag{A.8.2}$$

No caso geral, os elementos do grupo $\mathcal{G}_n$ são produtos tensoriais de n matrizes de Pauli com fatores multiplicativos ± 1 e $\pm i$. A menos do fator multiplicativo ± 1 ou $\pm i$, um elemento do grupo de Pauli $\mathcal{G}_n$ pode ser escrito como $Z_{\mathbf{v}}X_{\mathbf{w}}$ onde $\mathbf{v}$ e $\mathbf{w}$ são vetores binários com n componentes. Por exemplo, para $n = 3$, o elemento $Z \otimes Y \otimes X$ pode ser escrito como

$$\begin{aligned} Z \otimes Y \otimes X &= Z \otimes (-iZX) \otimes X \\ &= -i(Z \otimes Z \otimes I)(I \otimes X \otimes X) \\ &= -i\, Z_{(1,1,0)}X_{(0,1,1)}. \end{aligned}$$

Portanto, $\mathbf{v} = (1, 1, 0)$ e $\mathbf{w} = (0, 1, 1)$.

Um conjunto gerador do grupo de Pauli $\mathcal{G}_n$ é $\{Z_{\mathbf{e}_1}, \cdots, Z_{\mathbf{e}_n}, X_{\mathbf{e}_1}, \cdots, X_{\mathbf{e}_n}, i\, I^{\otimes n}\}$, onde $\mathbf{e}_j$ é o vetor binário onde somente a j-ésima componente tem valor 1. Um subgrupo particularmente útil de $\mathcal{G}_n$ é o grupo $\mathcal{K}_n = \langle Z_{\mathbf{e}_1}, \cdots, Z_{\mathbf{e}_n}, X_{\mathbf{e}_1}, \cdots, X_{\mathbf{e}_n} \rangle$. Todas as matrizes de $\mathcal{P}_n$ têm componentes reais. Os elementos de $\mathcal{P}_n$ podem ser expressos como produto tensorial de X, Z e iY sem o fator $\pm i$. O fator -1 está presente.

Outro grupo importante é o grupo quociente $\mathcal{G}_n/Z(\mathcal{G}_n)$, que é isomorfo ao grupo das *strings* binárias de comprimento $2n$ com soma binária bit-a-bit. O isomorfismo é obtido expressando os elementos de $\mathcal{G}_n/Z(\mathcal{G}_n)$ na forma $Z_{\mathbf{v}}X_{\mathbf{w}}$. Esse grupo também é isomorfo ao grupo dos *automorfismos internos* de $\mathcal{G}_n$.

Exercício A.14. *Seja S um subgrupo de $\mathcal{G}_n$. Mostre que $-I^{\otimes n} \notin S$ implica $\pm i\, I^{\otimes n} \notin S$.*

A.9 Grupo de Clifford

O *grupo de Clifford*, denotado por $\mathcal{C}_n$, é o normalizador de $\mathcal{G}_n$ em $U(2^n)$, isto é, é o grupo das matrizes unitárias U de dimensão 2^n que satisfazem $U\mathcal{G}_nU^\dagger = \mathcal{G}_n$. O *grupo de Clifford local*, denotado por $\mathcal{C}_n^l$, é o subgrupo de $\mathcal{C}_n$ de todas as matrizes que são produto tensorial de n matrizes do grupo $\mathcal{C}_1$. Todos o elementos do grupo de Pauli $\mathcal{G}_n$ pertencem ao grupo $\mathcal{C}_n^l$.

Para $n = 1$, o grupo de Clifford é gerado pelas matrizes *Hadamard* H e *fase* S, isto é, $\mathcal{C}_1 = \langle H, S \rangle$. $\mathcal{C}_1$ tem ordem 255. É simples verificar que H e S pertencem ao grupo de Clifford $\mathcal{C}_1$, pois

$$\begin{aligned} HXH &= Z \\ HYH &= -Y \\ HZH &= X \end{aligned}$$

e

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

Note que a conjugação de matrizes do grupo de Pauli por H ou S gera matrizes do grupo de Pauli.

Para $n \geq 2$, o grupo de Clifford é gerado pelas matrizes CNOT, *Hadamard* H e *fase* S, isto é,

$$\mathcal{C}_n = \langle H_1 \cdots H_n, S_1 \cdots S_n, \mathrm{CNOT}_{i,j}, \forall\, i \neq j \rangle.$$

$\mathcal{C}_n$ possui ordem $2^{n^2+2n+3} \prod_{j=1}^{n} (4^j - 1)$.

Sugestões para Leitura

A quantidade de bons livros de Teoria de Grupos é muito grande. Para um contato inicial, sugerimos as Refs. [13]; para uma abordagem mais avançada sugerimos a Ref. [9, 25, 24].

Bibliografia

[1] S. Beigi, I. Chuang, M. Grassl, P. Shor e B. Zeng. Graph concatenation for quantum codes. *Journal of Mathematical Physics*, **52** (2011), 022201.

[2] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin e W. K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, **54** (1996), 3824–3851.

[3] A. R. Calderbank, E. M. Rains, P. W. Shor e N. J. A. Sloane. Quantum error correction and orthogonal geometry. *Phys. Rev. Lett.*, **78** (1997), 405–408.

[4] A. R. Calderbank e P. W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, **54** (1996), 1098–1105.

[5] A. R. Calderbank, E. M. Rains, P. W. Shor e N. J. A. Sloane. Quantum error correction via codes over GF(4). *IEEE Transactions on Information Theory*, **44** (1998), 1369–1387.

[6] X. Chen, B. Zeng e I. L. Chuang. Nonbinary codeword-stabilized quantum codes. *Phys. Rev. A*, **78** (2008), 062315.

[7] I. Chuang, A. Cross, G. Smith, J. Smolin e B. Zeng. Codeword stabilized quantum codes: Algorithm and structure. *Journal of Mathematical Physics*, **50** (2009), 042109.

[8] A. Cross, G. Smith, J.A. Smolin e B. Zeng. Codeword stabilized quantum codes. *Information Theory, IEEE Transactions on*, (**55** (2009), 433–438.

[9] A. Garcia e Y.A.E. Lequain. "Elementos de Álgebra". Instituto de Matemática Pura e Aplicada - IMPA, 2002.

[10] D. Gottesman. "Stabilizer Codes and Quantum Error Correction". PhD thesis, Caltech, 1997.

[11] D. Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev. A*, **54** (1996), 1862–1868.

[12] M. Grassl, P. Shor, G. Smith, J. Smolin e B. Zeng. Generalized concatenated quantum codes. *Phys. Rev. A*, **79** (2009), 050306.

[13] I. N. Herstein. "Abstract algebra". Macmillan Pub., 1990.

[14] D. Hu, W. Tang, M. Zhao, Q. Chen, S. Yu e C. H. Oh. Graphical nonbinary quantum error-correcting codes. *Phys. Rev. A*, **78** (2008), 012306.

[15] P. Kaye, R. Laflamme e M. Mosca. "An Introduction to Quantum Computing". Oxford University Press, Inc., New York, NY, USA, 2007.

[16] E. Knill e R. Laflamme. Theory of quantum error-correcting codes. *Phys. Rev. A*, **55** (1997), 900–911.

[17] C. C. Lavor, M.M.S. Alves, R.M. Siqueira e S.I.R. Costa. "Uma Introdução à Teoria de Códigos", vol. 21 das *Notas em Matemática Aplicada*. Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), São Carlos, 2006.

[18] Y. Li, I. Dumer, M. Grassl e L. P. Pryadko. Structured error recovery for code-word-stabilized quantum codes. *Phys. Rev. A*, **81** (2010), 052337.

[19] N. Melo, D. F. G. Santiago e R. Portugal. Decoder for Nonbinary CWS Quantum Codes. *ArXiv e-prints:1204.2218*, 2012.

[20] M. A. Nielsen e I. L. Chuang. *Computação Quântica e Informação Quântica*. Editora Bookman, 2005.

[21] R. Portugal. "Algoritmos Quânticos de Busca", vol. 47 das *Notas em Matemática Aplicada*. Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), São Carlos, 2010.

[22] R. Portugal, C. C. Lavor, L. M. Carvalho e N. Maculan. "Uma Introdução à Computação Quântica", vol. 8 das *Notas em Matemática Aplicada*. Sociedade Brasileira de Matemática Aplicada e Computacional (SBMAC), São Carlos, 2004.

[23] P. W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, **52** (1995), R2493–R2496.

[24] K. Spindler. "Abstract Algebra with Applications: Rings and fields". M. Dekker, 1994.

[25] K. Spindler. "Abstract Algebra with Applications: Vector spaces and groups". M. Dekker, 1994.

[26] A. M. Steane. Error correcting codes in quantum theory. *Phys. Rev. Lett.*, **77** (1996), 793–797.

[27] A. M. Steane. Multiple particle interference and quantum error correction. *Proc. R. Soc. A*, **452** (1996), 2551.

[28] S. Yu, Q. Chen, C. H. Lai e C. H. Oh. Nonadditive quantum error-correcting code. *Phys. Rev. Lett.*, **101** (2008), 090501.

[29] S. Yu, Q. Chen e C. H. Oh. Graphical Quantum Error-Correcting Codes, September 2007. *ArXiv e-prints: 0709.1780*, 2007.

Índice