UNIVERSIDADE FEDERAL DE SANTA CATARINA
BIBLIOTECA UNIVERSITÁRIA

Graça Aparecida Prestes Sabadin

CÓDIGOS CORRETORES DE ERROS

Florianópolis

2019

Graça Aparecida Prestes Sabadin

CÓDIGOS CORRETORES DE ERROS

Dissertação submetida ao Programa de Mestrado Profissional em Matemática da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Matemática.
Orientador: Prof. Dr. Eduardo Tengan

Florianópolis

2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Sabadin, Graça Aparecida Prestes
Códigos Corretores de Erros / Graça Aparecida Prestes Sabadin ; orientador, Eduardo Tengan, 2019.
79 p.

Dissertação (mestrado profissional) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Programa de Pós Graduação em Matemática, Florianópolis, 2019.

Inclui referências.

1. Matemática. 2. Códigos Lineares. 3. Código de Hamming. 4. Códigos Cíclicos. 5. Código de Reed Solomon. I. Tengan, Eduardo. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Matemática. III. Título.

Graça Aparecida Prestes Sabadin
CÓDIGOS CORRETORES DE ERROS

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Banca Examinadora

Prof. Dr. Abdelmoubine Amar Henni
Universidade Federal de Santa Catarina.

Prof. Dr. Eliezer Batista
Universidade Federal de Santa Catarina.

Prof. Dr. Giuliano Boava
Universidade Federal de Santa Catarina.

Certificamos que esta é a versão original e final do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Matemática.

Prof. Dra. Maria Inez Cardoso Gonçalves
Coordenadora
Universidade Federal de Santa Catarina

Prof. Dr. Eduardo Tengan
Orientador
Universidade Federal de Santa Catarina

Florianópolis, 12 de setembro 2019 .

À minha família.

AGRADECIMENTOS

Ao meu orientador, Professor Dr. Eduardo Tengan, por ser um grande mestre, não só pelo seu saber, mas principalmente por sua generosidade e paciência, com que conduziu esse trabalho, a ele minha eterna gratidão e respeito.

Aos Professores da banca examinadora pelas contribuições.

Ao professor Dr. Celso Melchiades Doria, por acreditar e lutar pelo PROFMAT.

Aos professores do Profmat, pela dedicação na realização das aulas.

Aos colegas de mestrado, pela amizade e companheirismo.

Obrigada, Antônio, por teres feito minha inscrição no ENA. Obrigada, Waldir, pelo apoio ao emprestar-me todos os livros para a realização do curso. Obrigada, Valmiré e Angelita, pelo companheirismo nesta jornada. (Valmiré, valeu as inúmeras caronas.)

À CAPES, pelo apoio financeiro.

Queira
Basta ser sincero e desejar profundo
Você será capaz de sacudir o mundo, vai
Tente ou... tra vez
Tente
E não diga que a vitória está perdida
Se é de batalhas que se vive a vida
Tente outra vez

(Raul Seixas, 1975)

RESUMO

Este trabalho é dedicado ao estudo de códigos corretores de erros, os quais têm como base matemática a aritmética e a álgebra linear. Serão abordados os códigos lineares, tendo como exemplo o Código de Hamming, e uma subclasse dos códigos lineares que são os códigos cíclicos onde apresentaremos o Código de Reed-Solomon.

Palavras-chave: Códigos corretores de erros, Códigos Lineares: Código de Hamming, Código Cíclicos: Código de Reed-Solomon

ABSTRACT

This work is dedicated to the study of error correcting codes, which are mathematically based on arithmetic and linear algebra. Linear codes will be addressed, taking as an example the Hamming Code, as well as a subclass of linear codes, the cyclic ones, of which we shall present the Reed-Solomon Code.

Keywords: Error Correction Codes, Linear Codes: Hamming Code, Cyclic Code: Reed-Solomon Code

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Os códigos corretores de erros estão presentes em toda a parte no nosso cotidiano, como por exemplo: Wi-fi, CDs, DVDs, QRCode, transmissão de dados via satélite, telecomunicações, etc. A ideia básica da teoria é a codificação da informação enviada, a qual é feita adicionando informações redundantes à mensagem inicial, de modo que o receptor da mensagem consiga identificar possíveis erros causados pelo ruído na transmissão e desta forma possa recuperar a mensagem inicial.

Historicamente, a teoria teve início com os trabalhos de Shannon e Hamming por volta de 1950. Claude Elwood Shannon (1916–2001), matemático e nesta época funcionário da empresa Bell Labs, publicou em 1948 um artigo científico chamado "A mathematical theory of communication" no Bell System Journal. Este trabalho é considerado o marco inicial da Teoria de Informações, na qual a teoria dos códigos corretores tem papel importantíssimo. Neste artigo, Claude Shannon questiona sobre qual é a melhor maneira de codificar uma informação.

Figura 1 – Claude Shannon

Richard Wesley Hamming (1915–1998), matemático, funcionário da Bell Labs de 1946 a 1976, e colega de trabalho de Claude Shannon, investigou durante anos o problema de correção de erros e em 1950 publicou o algoritmo chamado "Hamming Code", o qual é utilizado em inúmeras áreas da computação. Outras importantes contribuições de Richard Hamming para a teoria dos códigos corretores de erros são: Distância de Hamming e Empacotamento de Esferas, assuntos que também serão abordadas neste trabalho. Em 1976, ele mudou-se para a Naval Postgraduate School, onde foi professor adjunto até 1997 quando se tornou professor emérito. No Youtube é possível assistir a inúmeras aulas do

Figura 2 – Richard Hamming

professor Richard Hamming, sendo uma delas sobre códigos corretores de erros. Assista em https://youtu.be/BZh07Ew32UA

Em seguida, faremos uma breve descrição dos tópicos abordados neste trabalho.

No capítulo 2, serão apresentados os conceitos, definições, termos utilizados e alguns exemplos da teoria básica de códigos corretores de erros.

Os códigos lineares são a classe de códigos mais utilizada e apresentaremos no capítulo 3 definições necessárias para o seu desenvolvimento. Em especial, trataremos de um dos principais códigos lineares, o código de Hamming, desenvolvido por Richard Hamming, o qual é utilizado amplamente nas telecomunicações, no processamento de sinal e nas memórias dos computadores, devido à sua simplicidade, pois permite a transferência e armazenamento de dados de forma eficiente e segura.

No Capítulo 4, veremos uma subclasse dos códigos lineares, os códigos cíclicos, e apresentaremos as definições necessárias para a sua criação. Dentre os inúmeros códigos cíclicos que existem estudaremos no capítulo seguinte o Código de Reed-Solomon (RS).

Os Códigos Reed-Solomon tiveram a importante aplicação de codificar as imagens digitais enviadas pela sonda espacial Voyager, em 1977, tornando-se difundida nas comunicações por satélite. Versões mais modernas foram e são utilizadas nas missões Mars Pathfinder, Galileo, Mars Exploration Rover e Cassini.

A codificação Reed-Solomon é amplamente utilizada em sistemas de armazenamento em massa para corrigir os erros de rajada associados a defeitos de mídia. Sendo um componente importante do CD, primeiro uso de uma forte codificação em produto de consumo produzido em larga escala. São utilizados também em códigos de barras bidimensionais, permitindo a leitura mesmo que uma parte do código de barras esteja danificada.

Para a decodificação e detecção de erros no código Reed-Solomon, utilizaremos o algoritmo de Peterson-Gorenstein-Zieler, desenvolvido por Daniel Gorenstein e Neal Zierler, o qual foi descrito em um relatório do MIT Lincoln Laboratory de Zierler em janeiro de 1960, e cuja descrição se encontra no livro "Error Correcting Codes", de W. Wesley Peterson (1961).

2 TEORIA DE CÓDIGOS

A situação geral considerada em Teoria de Códigos pode ser esquematizada na seguinte figura:

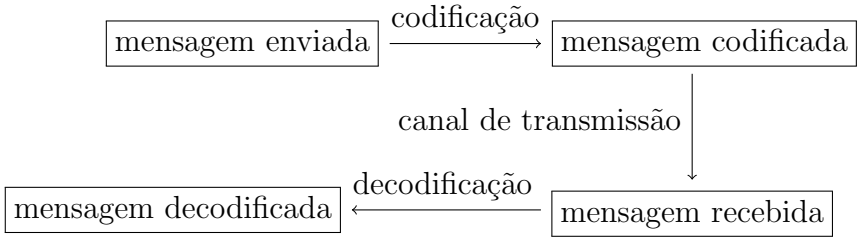

Figura 3 – Teoria Códigos

As mensagens codificadas e recebidas são palavras formadas por uma sequência de símbolos de um mesmo alfabeto. Para o envio da mensagem será utilizado um canal de transmissão que poderá ter ruído, desta forma a mensagem recebida poderá conter erros ou símbolos apagados e será diferente da mensagem enviada. O estudo dos códigos tem como objetivo encontrar códigos que enviem o maior número de palavras com maior rapidez, e que tenha capacidade de detectar e corrigir erros de modo eficiente.

Vamos definir alguns termos utilizados na teoria dos códigos.

Definição 2.0.1. (a) Um **alfabeto** é um conjunto finito de símbolos $\mathcal{A}_q = \{a_1, \ldots, a_q\}$.

(b) Uma **palavra** é uma sequência finita de elementos do alfabeto $\mathcal{A}_q$.

(c) Um **código q-ário** é um conjunto finito de palavras sobre um alfabeto de q elementos. Ao logo do texto nos referiremos a código q-ário como **código**.

(d) Se todas as palavras do código C têm o mesmo comprimento n, isto é, se $C \subset \mathcal{A}_q^n$, então C diz-se um **código uniforme**.

(e) Denotamos por $(n, M)_q$ um código uniforme q-ário com M palavras de comprimento n.

Um alfabeto pode ser qualquer conjunto finito de símbolos à nossa escolha. São também alfabetos os anéis $\mathbb{Z}/(m) = \{\overline{0}, \overline{1}, \ldots, \overline{m-1}\}$ (com $m \geq 2$ um número inteiro). Quando $\mathbb{Z}/(2) = \{\overline{0}, \overline{1}\}$ o código diz-se **binário**, e se $\mathbb{Z}/(3) = \{\overline{0}, \overline{1}, \overline{2}\}$ o código diz-se **ternário**. Note-se que $\mathbb{Z}/(2)$ e $\mathbb{Z}/(3)$ têm estrutura de corpo.

Utilizaremos a notação $x_1x_2\ldots x_n$ para a palavra $(x_1, x_2, \ldots, x_n) \in \mathcal{A}_q^n$, assim um vetor $(\overline{1}, \overline{0}, \overline{1}) \in (\mathbb{Z}/(2))^3$ será denotado por 101, e que também omitiremos as barras (se

não houver perigo de confusão).

O conjunto das letras $\mathcal{A} = \{a, b, c, \ldots, w, y, z\}$ é um alfabeto, e o conjunto de todas as palavras portuguesas formam um código P sobre $\mathcal{A}$ que não é uniforme. Nem todas as palavras no alfabeto $\mathcal{A}$ são palavras do código P; assim se recebermos uma palavra que não pertence a P, sabemos que ocorreu um erro na transmissão. Por exemplo, suponhamos que recebamos a palavra "calendáris", percebemos que ela não pertence a P, então o erro foi detectado e corrigiremos "calendáris" para a palavra mais próxima em P, que é "calendário". Isto motiva introduzirmos a seguinte definição de distância entre palavras:

Definição 2.0.2. Sejam $x = x_1x_2 \ldots x_n$, $y = y_1y_2 \ldots y_n \in \mathcal{A}_q^n$. Define-se a **distância de Hamming** entre as palavras x e y por:

$$d(x, y) = \#\{i : x_i \neq y_i\}.$$

Ou seja, $d(x, y)$ é o número de coordenadas em que x e y diferem, ou ainda, $d(x, y)$ é o número mínimo de trocas de símbolos necessárias para obter y a partir de x.

Definição 2.0.3. Seja C um código contendo pelo menos duas palavras. Define-se a **distância mínima** de C por:

$$d(C) = \min\{d(x, y) : x, y \in C, x \neq y\}.$$

Se C é um código q-ário com M palavras de comprimento n e distância mínima $d(C) = d$, dizemos que C é um código $(n, M, d)_q$. Os números n, M e d dizem-se **parâmetros** de C.

O parâmetro $d(C)$ será muito importante para discutirmos capacidade de detecção de erros de um código C.

Proposição 2.0.4. *A distância de Hamming é uma métrica, isto é, verifica as seguintes propriedades:*

(i) $d(x, y) \geq 0 \quad \forall x, y \in \mathcal{A}_q^n$;

(ii) $d(x, y) = 0 \Leftrightarrow x = y$;

(iii) simetria: $d(x, y) = d(y, x) \quad \forall x, y \in \mathcal{A}_q^n$;

(iv) desigualdade triangular: $d(x, y) \leq d(x, z) + d(z, y) \quad \forall x, y, z \in \mathcal{A}_q^n$.

Estas propriedades são consequência direta da definição de distância de Hamming.

Exemplo 2.0.5 (Distância de Hamming)**.** Definimos o seguinte código composto pelos vetores:

$$v_0 = (0,0,0) \qquad v_1 = (0,0,1) \qquad v_2 = (0,1,0) \qquad v_3 = (0,1,1)$$
$$v_4 = (1,0,0) \qquad v_5 = (1,0,1) \qquad v_6 = (1,1,0) \qquad v_7 = (1,1,1).$$

Temos que as distâncias entre os vetores são:

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_0	0	1	1	2	1	2	2	3
v_1	1	0	2	1	2	1	3	2
v_2	1	2	0	1	2	3	1	2
v_3	2	1	1	0	3	2	2	1
v_4	1	2	2	3	0	1	1	2
v_5	2	1	3	2	1	0	2	1
v_6	2	3	1	2	1	2	0	1
v_7	3	2	2	1	2	1	1	0

A distância deste código pode ser observada geometricamente no cubo, sendo cada um dos seus vértices um dos vetores. Para calcularmos a distância entre os vetores devemos verificar qual o menor número de arestas necessárias para conectá-los. Assim, por exemplo, os vetores que estão à distância 1 de v_0 são facilmente encontrados: v_1, v_2 e v_4.

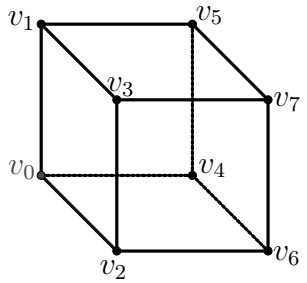

Figura 4 – Cubo

2.1 DECODIFICAÇÃO, DETECÇÃO E CORREÇÃO DE ERROS

A seguir, veremos maneiras de, encontrado um erro na mensagem enviada (ou seja, a palavra recebida não pertence ao código), determinar os casos em que conseguiremos corrigi-lo para a palavra mais próxima no código. A formulação precisa do problema é dada pela

Definição 2.1.1 (Decodificação por distância mínima)**.** Recebida a palavra $y \in \mathcal{A}_q^n$, procurar $x' \in C$ tal que

$$d(x', y) = \min\{d(x, y) : x \in C\}.$$

Por C ser finito, o conjunto $\{d(x, y) : x \in C\}$ também é finito e o mínimo na definição anterior sempre existe, embora, em geral, não haja um único $x' \in C$ que esteja a esta distância mínima de y.

Definição 2.1.2. Seja C um código e sejam s e t números inteiros positivos.

(a) Diz-se que C **detecta s erros** se e só se, quando ocorrem s erros ou menos, a palavra recebida não pertence ao código C.

(b) Diz-se que C **corrige t erros** se e só se o método de decodificação por distância mínima corrige t, ou menos, erros.

Exemplo 2.1.3. Na notação do Exemplo 2.0.5, considere as seguintes esferas, de acordo a figura abaixo:

$$\begin{aligned} S_1 &= \{x : d(x, v_1) \leq 1\} \\ &= \{v_1, v_0, v_5, v_3\}, \text{vetores que estão à distância no máximo 1 de } v_1. \end{aligned}$$

$$\begin{aligned} S_2 &= \{x : d(x, v_2) \leq 1\} \\ &= \{v_0, v_2, v_3, v_6\}, \text{vetores que estão à distância no máximo 1 de } v_2. \end{aligned}$$

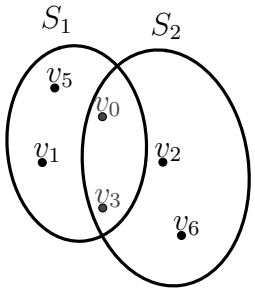

Figura 5 – Esferas S_1 e S_2

(a) Seja $C = \{001, 010\} = \{v_1, v_2\}$ código formado pelos vetores v_1 e v_2. Como $d(001, 010) = 2$, conclui-se que $d(C) = 2$ e portanto $(3, 2, 2)_2$ são os parâmetros deste código.

(b) Se a palavra 001 do código é enviada e ocorrem dois erros na transmissão dos símbolos e a palavra y recebida é 010, como $y \in C$ os erros não são detectados. Por outro lado, modificando uma letra em qualquer palavra em C obtemos palavras que não pertencem a C. Logo, C detecta um erro mas não detecta 2.

(c) Se uma palavra do código (001 ou 010) é enviada e a palavra recebida é $v_0 = 000$, concluímos que ocorreu um erro na transmissão pois v_0 não pertence ao código, mas não conseguimos corrigir o erro pelo fato de existir uma ambiguidade, visto que $v_0 \in S_1$ e $v_0 \in S_2$, conforme a figura. Logo, o código não corrige um erro.

(d) Entretanto, se uma palavra do código (001 ou 010) é enviada e a palavra recebida é $v_5 = 101$, concluímos que ocorreu um erro na transmissão pois v_5 não pertence ao código, e a palavra é decodificada por $v_1 = 001$, pois é a **única** palavra de C que está à distância no máximo 1 de v_5.

Teorema 2.1.4. *Seja C um código com distância mínima $d(C)$. Então:*

(a) C detecta s erros se e só se $d(C) \geq s + 1$;

(b) C corrige t erros se e só se $d(C) \geq 2t + 1$.

Demonstração. (a) Suponhamos que $d(C) \geq s + 1$. Seja $x \in C$ a palavra enviada e suponhamos que ocorrem no máximo s erros na transmissão e $y \neq x$ é a palavra recebida. Portanto $0 < d(x, y) \leq s$. Como $0 < d(x, y) < d(C)$, conclui-se que $y \notin C$ e os s erros são detectados. Reciprocamente, se $d(C) \leq s$, então existem palavras $x, y \in C$ tais que $d(x, y) = d(C) \leq s$. Logo é possível x ser a palavra enviada, ocorrerem $d(C)$ erros e recebermos a palavra y. Como $y \in C$, estes erros não são detectados (observe o item (b) do Exemplo 2.1.3).

(b) ($\Longleftarrow$) Suponhamos que $d(C) \geq 2t+1$. Seja $x \in C$ a palavra enviada e suponhamos que ocorrem t erros na transmissão e $y \neq x$ é a palavra recebida. Portanto $0 < d(x, y) \leq t$. Para qualquer $c \in C$, com $c \neq x$, temos

$$d(x, c) \leq d(x, y) + d(y, c) \tag{2.1}$$

logo

$$d(y, c) \geq d(x, c) - d(x, y) \geq d(C) - t \geq 2t + 1 - t = t + 1 > d(x, y), \tag{2.2}$$

e assim, usando o método de decodificação por distância mínima, y é decodificado corretamente por x (observe o item (d) do Exemplo 2.1.3).

(b) ($\Longrightarrow$) Seja C um código que corrige t erros e suponhamos por absurdo $d = d(C) \leq 2t$. Então existem $x, x' \in C$ tais que $d(x, x') = d(C) \leq 2t$.

Sem perda de generalidade, podemos também assumir que x e x' diferem precisamente nas primeiras $d = d(C)$ coordenadas. Seja

$$y = \underbrace{x_1 \cdots x_t}_{\text{como } x'} \underbrace{x_{t+1} \cdots x_d}_{\text{como } x} \underbrace{x_{d+1} \cdots x_n}_{\text{como } x \text{ e } x'} \tag{2.3}$$

a palavra recebida. Temos $d(y, x') = d - t \leq t = d(y, x)$. Há dois casos a considerar.

- Se $d(y, x') < d(y, x)$ e x é a palavra transmitida então y é decodificada incorretamente por x'.
- Se $d(y, x') = d(y, x)$, não podemos decidir entre x e x' na decodificação por distância mínima.

(observe o item (c) do Exemplo 2.1.3).

□

Corolário 2.1.5. *Seja C um código de distância mínima $d(C) = d$. Então C detecta precisamente $d - 1$ erros e corrige precisamente $\left\lfloor \frac{d-1}{2} \right\rfloor$ erros.*

2.2 O PROBLEMA PRINCIPAL DA TEORIA DE CÓDIGOS

O principal problema na Teoria de Códigos é determinar o maior número de palavras possível que um código q-ário de comprimento n e distância mínima d pode conter (a fim de enviar mais mensagens, corrigir mais erros e ter uma taxa de transmissão maior). Ou seja, para q, n e d fixos, determinar

$$A_q(n, d) := \max\{M : \exists \text{ código } q\text{-ário } (n, M, d)\}. \tag{2.4}$$

Determinar $A_q(n, d)$ para parâmetros n e d arbitrários é um problema extremamente difícil e conhecem-se poucos resultados concretos. Faremos no lugar uma estimativa.

Usando a distância de Hamming d, se $c \in \mathcal{A}_q^n$ e r é um inteiro não negativo, a bola (ou esfera) de centro c e raio r é o subconjunto de $\mathcal{A}_q^n$ definido por:

$$B_r(c) = \left\{ x \in \mathcal{A}_q^n : d(x, c) \leq r \right\}.$$

Sendo $\mathcal{A}_q$ um conjunto finito, $\mathcal{A}_q^n$ e qualquer subconjunto deste também o são.

Define-se o **volume** de um subconjunto S de $\mathcal{A}_q^n$ por:

$$\text{vol}(S) = \#S.$$

Exemplo 2.2.1 (Volume de $\boldsymbol{B_2(00000) \subset \mathcal{A}_4^5}$)**.** Neste caso o centro da bola é o vetor 00000 e o raio é dois, queremos calcular o volume do conjunto formado pelos vetores que estão à distância no máximo 2 do centro da bola.

O único vetor que está à distância zero do centro é o próprio centro 00000.

Os vetores que estão à distância 1 do centro são todos os que diferem em exatamente uma coordenada do vetor 00000, como, por exemplo, 10000, 01000, 00100, ..., 20000, 02000, ..., 00030, 00003. Para encontrá-los, há $\binom{5}{1} = 5$ maneiras de escolha de posição do dígito distinto, que pode ser qualquer um dos $4-1=3$ valores $1, 2, 3$. Logo, o número destes vetores é $5 \cdot 3 = 15$.

Os vetores que estão à distância 2 do centro são todos que diferem em exatamente duas coordenadas do centro, por exemplo, 11000, 02300, ... Podemos escolher as posições destas coordenadas de $\binom{5}{2} = 10$ maneiras; daí basta multiplicar por $3^2 = 9$, pois temos 3 possibilidades para uma das coordenadas e 3 possibilidades para a outra.

O volume é dado pela quantidade total dos vetores que estão à distância 0, 1 e 2 do centro da bola. Assim,

$$\begin{aligned}\text{vol}\left(B_2(00000)\right) &= \binom{5}{0}(4-1)^0 + \binom{5}{1}(4-1)^1 + \binom{5}{2}(4-1)^2 \\ &= 1 + 15 + 90 = 106\end{aligned}$$

Geometricamente, podemos pensar na bola como formada por "camadas de cebola", conforme figura abaixo, sendo c o centro da bola:

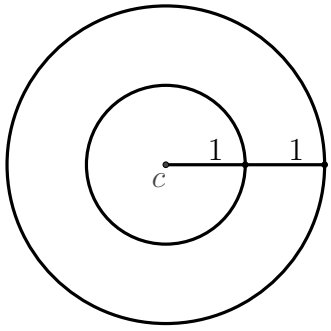

Figura 6 – Volume da bola ("camadas de cebola")

No caso geral, temos

Lema 2.2.2. *O volume da bola $B_r(c)$ é*

$$\mathrm{vol}\left(B_r(c)\right) = \sum_{j=0}^{r} \binom{n}{j} (q-1)^j$$

onde $0 \leq r \leq n$ e $c \in \mathcal{A}_q^n$.

Demonstração. A bola $B_r(c)$ é a união disjunta dos conjuntos

$$\left\{x \in \mathcal{A}_q^n : d(x,c) = j\right\}$$

com $j = 0, 1, \ldots, r$. Portanto

$$\mathrm{vol}\left(B_r(c)\right) = \sum_{j=0}^{r} \#\left\{x \in \mathcal{A}_q^n : d(x,c) = j\right\}.$$

Seja $c = (c_1, \ldots, c_n)$. Como

- $d(x,c) = j$ se e somente se x e c diferem exatamente em j coordenadas,
- $\binom{n}{j}$ é o número de maneiras diferentes de escolher j coordenadas em n e
- $q-1$ é o número de símbolos em $\mathcal{A}_q \backslash \{c_i\}$, isto é, o número de escolhas para a coordenada $x_i \neq c_i$,

conclui-se que

$$\#\left\{x \in \mathcal{A}_q^n : d(x,c) = j\right\} = \binom{n}{j}(q-1)^j.$$

Caso $r \geq n$ tem-se obviamente que $B_r(c) = \mathcal{A}_q^n$, cujo volume é q^n. □

Agora podemos apresentar uma cota superior para o número de palavras de um código.

Teorema 2.2.3 (Estimativa de Hamming ou Majorante de Empacotamento de Esferas)**.** *Para $q \geq 2$ e $2t+1 \leq d \leq n$, sendo t o número de erros que o código corrige, temos*

$$A_q(n,d) \leq \frac{q^n}{\mathrm{vol}\left(B_t(c)\right)}.$$

Demonstração. Seja C um código $(n, M, d)_q$ com $M = A_q(n,d)$ e $d \geq 2t+1$. Assim, as M bolas de raio t e centro nas M palavras do código são disjuntas duas a duas. Então

$$\mathrm{vol}\left(\bigcup_{c \in C} B_t(c)\right) = \sum_{c \in C} \mathrm{vol}\left(B_t(c)\right) = M\,\mathrm{vol}\left(B_t(c)\right)$$

uma vez que as bolas com o mesmo raio têm volumes iguais. Como $\text{vol}\left(\mathcal{A}_q^n\right) = q^n$, a igualdade acima implica que $M \,\text{vol}\left(B_t(c)\right) \leq q^n$. □

Exemplo 2.2.4. Seja $C_1 = \{000, 111\} = \{v_0, v_7\}$ na notação do Exemplo 2.0.5. A distância mínima é $d(C_1) = 3$, o número de palavras é $M = 2$ e como $d \geq 2t + 1$, temos que o C_1 corrige $t = 1$ erro.

O $\text{vol}\left(B_1(000)\right) = 4$, então

$$\text{vol}\left(\bigcup_{c \in C} B_1(c)\right) = M \,\text{vol}\left(B_1(c)\right) = 2 \cdot 4 = 8 \leq 2^3.$$

Observe a figura abaixo:

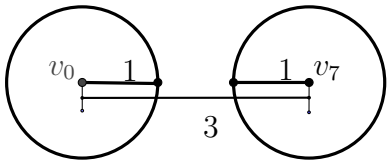

Figura 7 – Empacotamento de Esferas

3 CÓDIGOS LINEARES

Códigos Lineares formam a classe de códigos mais utilizada. Denotando por $\mathbb{F}_q$ o corpo com q elementos, temos

Definição 3.0.1. Um **código linear** q-ário de comprimento n é um subespaço vetorial de $\mathbb{F}_q^n$.

Seja k a dimensão do código C e seja $v_1, v_2, \ldots, v_k$ uma de suas bases. Todo elemento de C se escreve de modo único da seguinte forma: $\alpha_1 v_1 + \cdots + \alpha_k v_k$, onde α_i são elementos de $\mathbb{F}_q$ para $i = 1, \ldots, k$. Assim, $M = |C| = q^k$. Resumindo

Proposição 3.0.2. *Seja C um código linear de comprimento n sobre $\mathbb{F}_q$. Então: $|C| = q^{\dim C}$, isto é, $\dim C = \log_q |C|$.*

Como o número de palavras que C contém está diretamente relacionado com a sua dimensão, definimos os parâmetros de um código linear como sendo $(n, k, d)_q$ (ou simplesmente (n, k, d), ou ainda (n, k)), onde $k = \dim C$, e n e d são respectivamente o comprimento e a distância mínima. Portanto, um código linear $(n, k, d)_q$ é também um código $(n, q^k, d)_q$.

Um código linear contém necessariamente o vetor nulo.

Definição 3.0.3. Seja C um código linear. Definimos o **peso** $w(x)$ de $x \in C$ como sendo o número de entradas não nulas do vetor x, ou seja, $w(x) = d(x, \overrightarrow{0})$. Definimos o **peso mínimo** de C por:

$$w(C) = \min\{w(x) : x \in C \backslash \{\overrightarrow{0}\}\},$$

se $C \neq \{\overrightarrow{0}\}$, e $w(C) = 0$ se $C = \{\overrightarrow{0}\}$.

Veremos que podemos calcular a distância mínima de um código linear C utilizando o peso mínimo.

Teorema 3.0.4. *Seja $C \neq \{\overrightarrow{0}\}$ um código linear. Então*

$$d(C) = w(C).$$

Demonstração. Como $C \neq \{\overrightarrow{0}\}$, C contém pelo menos duas palavras e, de acordo com a definição de distância mínima, $d(C) = \min\{d(x, y) : x, y \in C, x \neq y\}$. Sejam então $x, y \in C$ tais que $d(x, y) = d(C)$. Portanto:

$$d(C) = d(x, y) = w(x - y) \geq w(C).$$

Na desigualdade usou-se o fato de $x - y \in C$, por C ser linear, e $x - y \neq \overrightarrow{0}$. Seja agora $x \in C$ tal que $w(x) = w(C)$. Portanto

$$w(C) = w(x) = d(x, \overrightarrow{0}) \geq d(C).$$

Na desigualdade usou-se o fato de $\overrightarrow{0} \in C$, por C ser linear. □

Por definição, para calcular a distância mínima $d(C)$ de um código qualquer C contendo M palavras é preciso calcular a distância $d(x, y)$ para $\binom{M}{2} = \frac{M(M-1)}{2}$ pares de palavras. Se C é linear, o teorema anterior diz-nos que basta calcular o peso $w(x)$ de $M-1$ palavras.

A seguir veremos como codificar uma mensagem para enviá-la e em seguida como verificar se a mensagem recebida pertence ao código.

3.1 MATRIZ GERADORA E MATRIZ DE PARIDADE

Em todo o texto vamos escrever vetores no formato de vetores colunas.

Definição 3.1.1. Seja C um código linear q-ário (n, k).

- Seja $\beta = \{v_1, \ldots, v_k\}$ uma base ordenada de C. Considere a matriz G, cujas colunas são os vetores $v_i = (v_{i1}, \ldots, v_{in})$, $i = 1, \ldots, k$, isto é,

$$G = \begin{bmatrix} | & & | \\ v_1 & \ldots & v_k \\ | & & | \end{bmatrix}.$$

  Dizemos que G é uma **matriz geradora** de C.

- Suponha que C seja o núcleo de uma transformação linear sobrejetora $H \colon \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$. A **matriz de paridade** de C é a matriz de H nas bases canônicas de $\mathbb{F}_q^n$ e $\mathbb{F}_q^{n-k}$.

Acima, como $H \colon \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$ é sobrejetora, então, pelo teorema do núcleo e da imagem, temos que

$$\dim \mathbb{F}_q^n = \dim(\ker(H)) + \dim(\mathrm{im}(H))$$

$$\iff n = \dim C + n - k$$

$$\iff \dim C = k.$$

Para existir uma matriz geradora devemos ter $\dim C > 0$ e para existir uma matriz de paridade devemos ter $\dim C < n$.

Uma matriz geradora tem n linhas e k colunas, e uma matriz de paridade tem $n-k$ linhas e n colunas.

Seja C um código linear de dimensão k e comprimento n, e seja G a matriz geradora de C, associada à base β. O código será a imagem da transformação linear injetiva definida por:

$$\begin{aligned} T\colon \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ x &\longmapsto Gx. \end{aligned}$$

Se $x = \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}$, temos que

$$T(x) = Gx = x_1v_1 + x_2v_2 + \cdots + x_kv_k,$$

Então, $T(\mathbb{F}_q^k) = C$, assim, qualquer vetor $x \in \mathbb{F}_q^k$ (mensagem que será enviada) é codificada por T, transformando-se numa sequência chamada de **código do canal**, que é o resultado da transformação.

Uma matriz de paridade $H_{(n-k)\times n}$ de um código linear C com matriz geradora G é uma matriz que detecta se um vetor y de $\mathbb{F}_q^n$ é uma palavra do código. Assim,

$$Hy = 0 \in \mathbb{F}_q^{n-k} \Longleftrightarrow y \in C \subset \mathbb{F}_q^n.$$

A partir disto, temos que, dado $y \in \mathbb{F}_q^n$, a imagem deste elemento pela matriz de paridade, $Hy \in \mathbb{F}_q^{n-k}$, é chamada **sintoma** de y. Logo, os elementos que pertencem ao código C têm por sintoma o vetor nulo. No caso de $Hy \neq 0$, assumimos que ocorreram erros na transmissão de y, pois neste caso $y \notin C$.

Recebido $y \in C$, para encontrar o vetor $x \in \mathbb{F}_q^k$, que foi codificado por T, devemos resolver o sistema $Gx = y$, o qual pode ser bem trabalhoso se a matriz geradora não estiver na forma canônica (o que veremos em breve na seção 3.4).

A seguir, veremos como exemplo o Código de Hamming, um código especial por ser **perfeito** (i.e., um código que atinge a cota máxima dada no teorema 2.2.3). Isto que possibilita uma taxa mais alta de transmissão para códigos de seu comprimento e ainda possui distância mínima 3, o que possibilita a detecção de até dois erros e a correção de um.

3.2 EXEMPLO: CÓDIGOS DE HAMMING

Seja H uma matriz cujas colunas são todos os vetores não nulos do espaço vetorial $\mathbb{F}_2^r$. Portanto H tem r linhas e $2^r - 1$ colunas. Além disso, como os vetores da base canônica $\overrightarrow{e_1}, \ldots, \overrightarrow{e_r}$ são colunas de H, a matriz identidade I_r é uma submatriz de H com determinante $\det(I_r) = 1 \neq 0$, portanto, as r linhas de H são linearmente independentes, e H é uma matriz de paridade de um código binário.

Definição 3.2.1. Seja H uma matriz $r \times (2^r - 1)$ cujas colunas são todos vetores em $\mathbb{F}_2^r \setminus \{\overrightarrow{0}\}$. O código binário $\mathrm{Ham}(r, 2)$ com esta matriz de paridade H diz-se um **código de Hamming** binário de redundância r.

Lema 3.2.2. *Seja $r \geq 2$. A distância mínima do código de Hamming é 3. Então* $\mathrm{Ham}(r, 2)$ *tem parâmetros* $(2^r - 1, 2^r - r - 1, 3)$.

Demonstração. Por construção, $\mathrm{Ham}(r, 2)$ tem comprimento $|\mathbb{F}_2^r \setminus \{\overrightarrow{0}\}| = 2r - 1$ e dimensão $k = n - r = 2r - r - 1$. Só falta ver que a distâcia mínima é $d = 3$. Sejam c_i, com $i = 1, \cdots, 2r - 1$, as colunas de uma matriz de paridade H para $\mathrm{Ham}(r, 2)$. Por construção, $c_i \neq c_j$ para quaisquer $i \neq j$, e nenhuma coluna é o vector nulo, logo quaisquer duas colunas de H são linearmente independentes. Por outro lado $c_i = (0, ..., 0, 0, 1)$, $c_j = (0, ..., 0, 1, 0)$ e $c_k = (0, ..., 0, 1, 1)$ são colunas de H, se $r \geq 2$. Como $c_k = c_i + c_j$, estas três colunas são linearmente dependentes. Logo, pelo Teorema 3.5.1 $d(\mathrm{Ham}(r, 2)) = 3$. □

Exemplo 3.2.3. Para $r = 2$, $\mathrm{Ham}(2, 2)$ tem parâmetros $(3, 1, 3)$. Para $r = 3$, $\mathrm{Ham}(3, 2)$ tem parâmetros $(7, 4, 3)$.

Exemplo 3.2.4. Vamos ver o Código de Hamming $\mathrm{Ham}(3, 2)$.

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

Esta é a matriz paridade de um código $\mathrm{Ham}(3, 2)$. Para facilitar a decodificação as colunas da matriz contém a representação binária dos números 1 a $2^3 - 1$, os quais estão ordenadas em ordem crescente.

Assim, o código binário $\mathrm{Ham}(3, 2)$ é dado pelas palavras-código que são soluções do

sistema linear

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.1}$$

Logo, temos o seguinte sistema linear:

$$\begin{cases} & & & & x_4+ & x_5+ & x_6+ & x_7 = 0 \\ & x_2+ & x_3+ & & & & x_6+ & x_7 = 0 \\ x_1 & + & x_3 & + & x_5 & & + & x_7 = 0 \end{cases} \tag{3.2}$$

Resolvendo o sistema temos:

$$\begin{cases} x_1 = x_3 + x_5 + x_7 \\ x_2 = x_3 + x_6 + x_7 \\ x_4 = x_5 + x_6 + x_7 \end{cases} \tag{3.3}$$

O sistema apresenta 4 variáveis independentes (x_3, x_5, x_6 e x_7), que se referem ao código da fonte, e 3 variáveis dependentes (x_1, x_2 e x_4), que representam a redundância acrescentada, o que torna possível detectar e corrigir possíveis erros. Assim, o código $\text{Ham}(3, 2)$ transforma uma palavra-código de 4 dígitos em uma palavra-código de 7 dígitos com o acréscimo de uma redundância de 3 dígitos.

Para encontrarmos a matriz G, geradora do código, note que temos o seguinte sistema:

$$\begin{cases} x_1 = 1x_3 + 1x_5 + 0x_6 + 1x_7 \\ x_2 = 1x_3 + 0x_5 + 1x_6 + 1x_7 \\ x_3 = 1x_3 + 0x_5 + 0x_6 + 0x_7 \\ x_4 = 0x_3 + 1x_5 + 1x_6 + 1x_7 \\ x_5 = 0x_3 + 1x_5 + 0x_6 + 0x_7 \\ x_6 = 0x_3 + 0x_5 + 1x_6 + 0x_7 \\ x_7 = 0x_3 + 0x_5 + 0x_6 + 1x_7 \end{cases} \tag{3.4}$$

O sistema equivale ao seguinte produto de matrizes:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \tag{3.5}$$

Logo, a matriz geradora é dada por

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

Esta matriz geradora G gera 16 palavras-códigos como veremos a seguir:

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.6}$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.7}$$

$$\vdots$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{3.8}$$

Na tabela abaixo encontramos todas estas 16 palavras-códigos.

$x_3x_5x_6x_7$	$x_3+x_5+x_7, x_3+x_6+x_7, x_3, x_5+x_6+x_7, x_5, x_6, x_7$
0 0 0 0	0 0 **0** 0 **0 0 0**
0 0 0 1	1 1 **0** 1 **0 0 1**
0 0 1 0	0 1 **0** 1 **0 1 0**
0 0 1 1	1 0 **0** 0 **0 1 1**
0 1 0 0	1 0 **0** 1 **1 0 0**
0 1 0 1	0 1 **0** 0 **1 0 1**
0 1 1 0	1 1 **0** 0 **1 1 0**
0 1 1 1	0 0 **0** 1 **1 1 1**
1 0 0 0	1 1 **1** 0 **0 0 0**
1 0 0 1	0 0 **1** 1 **0 0 1**
1 0 1 0	1 0 **1** 1 **0 1 0**
1 0 1 1	0 1 **1** 0 **0 1 1**
1 1 0 0	0 1 **1** 1 **1 0 0**
1 1 0 1	1 0 **1** 0 **1 0 1**
1 1 1 0	0 0 **1** 0 **1 1 0**
1 1 1 1	1 1 **1** 1 **1 1 1**

3.3 ALGORITMO DE DECODIFICAÇÃO PARA OS CÓDIGOS DE HAMMING BINÁRIOS:

Suponha que as colunas de H estão ordenadas por ordem crescente, isto é, a i-ésima coluna é o número $i \in \{1, \ldots, n = 2^r - 1\}$ escrito na base 2, logo se $e_i = (0, \ldots, 0, 1, 0 \ldots, 0)$ com 1 na coordenada i, o sintoma $S(e_i)$ é a representação binária de i. Assim, temos o seguinte algoritmo de decodificação para $\mathrm{Ham}(r, 2)$:

Este algoritmo corrige no máximo um erro. Ver Lema 3.2.2.

(1) Recebido $y \in \mathbb{F}_2^n$, calcular o sintoma $S(y) = Hy$.

(2) Se $S(y) = 0$, assumir que não ocorreram erros de transmissão e decodificar y por y.

(3) Se $S(y) \neq 0$, então $S(y)$ é uma coluna de H e, se estas estão por ordem crescente, assumir que ocorreu um erro na coordenada i correspondente ao número $S(y)$ na base 2, e decodificar y por $y - e_i$.

Por que o algoritmo funciona?

Suponha que a palavra y é enviada e a mensagem r é recebida e que ocorreu um erro na i-ésima coordenada de y, foi trocado um por zero ou zero por um.

Então,

$$r = y + e_i,$$

o vetor e_i possui zeros em todas as coordenadas exceto na i-ésima posição. Observemos que saber a posição do erro é suficiente para determinar a mensagem y, pois

$$y = r - e_i,$$

visto que os códigos são binários. Desta forma,

$$H \cdot r = H \cdot (y + e_i) = H \cdot y + H \cdot e_i$$

Como y é uma palavra do código, temos que $H \cdot y = 0$. Então,

$$H \cdot r = H \cdot e_i$$

Assim,

$$H \cdot e_i = H \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Concluímos que He_i é a i-ésima coluna da matriz H, localizamos o erro e decodificamos y.

Exemplo 3.3.1. Seja $C = \mathrm{Ham}(3, 2)$. Supondo que recebemos o vetor $y = 1111010$, com $y \in \mathbb{F}_2^7$, calcularemos o sintoma $S(y) = H \cdot y$.

$$S(y) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Encontramos o sintoma $S(y) \neq 0$, de acordo com o algoritmo de decodificação, $S(y)$ é uma coluna de H, as quais estão em ordem crescente, assim podemos assumir que ocorreu um erro na coordenada i correspondente ao número $S(y)$ na base 2. Neste caso assumimos que ocorreu um erro na coordenada 2, pois $(010)_2 = (2)_{10}$ e decodificamos y por $y - e_2$.

Assim,

$$y = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

E o vetor é decodificado corretamente por 1011010.

Para encontrarmos qual o vetor que foi enviado precisamos resolver o sistema $Gx = y$:

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_3 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{3.9}$$

$$\begin{cases} 1x_3 + 1x_5 + 0x_6 + 1x_7 = 1 \\ 1x_3 + 0x_5 + 1x_6 + 1x_7 = 0 \\ 1x_3 + 0x_5 + 0x_6 + 0x_7 = 1 \\ 0x_3 + 1x_5 + 1x_6 + 1x_7 = 1 \\ 0x_3 + 1x_5 + 0x_6 + 0x_7 = 0 \\ 0x_3 + 0x_5 + 1x_6 + 0x_7 = 1 \\ 0x_3 + 0x_5 + 0x_6 + 1x_7 = 0 \end{cases} \tag{3.10}$$

e então encontramos $x_3 = 1$, $x_5 = 0$, $x_6 = 1$ e $x_7 = 0$, assim, o vetor (palavra) enviado(a) foi 1010.

3.4 MATRIZ GERADORA E MATRIZ DE PARIDADE NA FORMA CANÔNICA

Como a matriz G geradora de um código linear C não é única, pois depende da escolha da base β, temos que duas matrizes geradoras de um mesmo código C podem ser obtidas uma da outra por uma sequência de operações. Então, de acordo com o estudo de álgebra linear, são permitidas as seguintes operações:

(a) Permutação de duas colunas.

(b) Multiplicação de uma coluna por um escalar não nulo.

(c) Adição de um múltiplo escalar de uma coluna a outra.

Podemos também permutar as coordenadas, ou seja, permutar as linhas, obtendo um código isomorfo ao original.

Desta forma, podemos sempre encontrar uma matriz geradora na forma canônica, isto é, de modo que as primeiras k linhas de G formem uma matriz identidade de ordem k,

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

Assim, se tivermos um código definido por uma matriz geradora G que não está na forma canônica, poderemos efetuar operações elementares por colunas nesta até obter uma matriz na forma canônica. O subespaço gerado pelas colunas da matriz padrão obtida é o mesmo e, portanto, temos o mesmo código.

Estando a matriz geradora na forma canônica, a codificação terá a seguinte forma:

$$T(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ c_1 \\ c_2 \\ \vdots \\ c_{n-k} \end{bmatrix},$$

onde os c_i's são combinações lineares das primeiras coordenadas. Neste caso, $(x_1, \ldots, x_k)$ são os **dígitos da mensagem** e $(c_1, \ldots, c_{n-k})$ são os **dígitos de verificação** ou **de redundância**. Dado um vetor codificado $y \in C$, a mensagem x é obtida simplesmente apagando os dígitos de verificação.

Se G é uma matriz geradora na forma canônica, então uma matriz de paridade para C será

$$H = \begin{bmatrix} -B & I_{(n-k)} \end{bmatrix},$$

Se o código for binário, isto é, sobre o alfabeto $\mathbb{F}_2$, teremos $B = -B$.

Exemplo 3.4.1. Seja $C \subset \mathbb{F}_7^6$ gerado pelo conjunto $S = \{122100, 012210, 001221\}$.

Vamos determinar uma matriz geradora e uma matriz paridade para C. Seja M a matriz cujas colunas são os vetores do conjunto S, aplicaremos o método de eliminação de Gauss a M.

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 2 & 1 \\ 1 & 2 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \xrightarrow{c_2 \mapsto 5c_3 + c_2} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 5 & 2 \\ 0 & 4 & 2 \\ 0 & 5 & 1 \end{bmatrix} \xrightarrow{c_1 \mapsto 5c_2 + c_1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \\ 5 & 5 & 2 \\ 6 & 4 & 2 \\ 4 & 5 & 1 \end{bmatrix} \xrightarrow{c_1 \mapsto 5c_3 + c_1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 5 & 2 \\ 2 & 4 & 2 \\ 2 & 5 & 1 \end{bmatrix} = \bar{M}.$$

Como $\bar{M}$ foi obtida de M aplicando apenas operações nas colunas, $\bar{M}$ e M têm o mesmo espaço gerado pelas colunas, assim, a matriz geradora G na forma canônica para

o código C é:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 5 & 2 \\ 2 & 4 & 2 \\ 2 & 5 & 1 \end{bmatrix}.$$

Como a matriz B é

$$B = \begin{bmatrix} 1 & 5 & 2 \\ 2 & 4 & 2 \\ 2 & 5 & 1 \end{bmatrix}$$

e para a matriz paridade, precisamos de $-B$, temos que

$$-B = \begin{bmatrix} -1 & -5 & -2 \\ -2 & -4 & -2 \\ -2 & -5 & -1 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 5 \\ 5 & 3 & 5 \\ 5 & 2 & 6 \end{bmatrix},$$

pois $C \subseteq \mathbb{F}_7^6$, então, a matriz de paridade H na forma canônica é:

$$H = \begin{bmatrix} 6 & 2 & 5 & 1 & 0 & 0 \\ 5 & 3 & 5 & 0 & 1 & 0 \\ 5 & 2 & 6 & 0 & 0 & 1 \end{bmatrix}.$$

Assim, para enviarmos a mensagem $x = (101)$, iremos codificá-la utilizando a matriz geradora G.

$$Gx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 5 & 2 \\ 2 & 4 & 2 \\ 2 & 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 3 \\ 4 \\ 3 \end{bmatrix} = y.$$

Recebido o vetor y, calculamos o sintoma $S(y) = Hy$ e encontramos que $Hy = 0$, logo $y \in C$ e para encontrarmos a mensagem enviada basta apagar os dígitos de redundância, últimos três dígitos do vetor.

3.5 DISTÂNCIA MÍNIMA

Teorema 3.5.1. *Seja C um código (n, k) sobre $\mathbb{F}_q$, com matriz de paridade H. Então*

(i) $d(C) \geq d$ *se, e somente se quaisquer* $d-1$ *colunas de* H *são linearmente independentes,*

(ii) $d(C) \leq d$ *se, e somente se existem* d *colunas de* H *linearmente dependentes.*

Demonstração. Pelo Teorema 3.0.4, sabemos que $d(C) = w(C)$. Designemos $c_1, \ldots, c_n$ as colunas da matriz paridade de H. Seja $x = (x_1, \ldots, x_n)$ uma palavra do código $C \in \mathbb{F}_q^n$ com peso $w(x) = e > 0$ e suponhamos que as componentes de x não nulas se encontram nas coordenadas $i_1, \ldots, i_e$. Como $C = \ker(H)$, temos:

$$x \in C \iff Hx = \overrightarrow{0} \iff \sum_{i=1}^{n} x_i c_i = \overrightarrow{0}$$

$$\iff x_{i_1} c_{i_1} + \cdots + x_{i_e} c_{i_e} = \overrightarrow{0} \quad \text{com } x_{i_1}, \ldots, x_{i_e} \neq 0$$

$$\iff \text{existem } e = w(x) \text{ colunas de } H \text{ linearmente dependentes.}$$

(i) Por definição de peso mínimo, $w(C) \geq d$ se, e somente se $w(x) \geq d$ para todas as palavras de código $x \in C \backslash \{\overrightarrow{0}\}$, ou seja, se e somente se C não contém nenhuma palavra x não nula com peso $w(x) \leq d-1$. Esta última afirmação equivale a dizer que quaisquer $d-1$ colunas de H são linearmente independentes.

(ii) Analogamente à alínea (i), $w(C) \leq d$ se, e somente se existe uma palavra não nula x do código C com $0 < w(x) \leq d$, o que é equivalente a existir um conjunto linearmente dependente de d colunas de H.

□

Juntando as duas afirmações deste teorema, podemos dizer que a distância mínima de um código linear C com a matriz de paridade H é dada por:

$$\boxed{d(C) = \text{número mínimo de colunas de } H \text{ linearmente dependentes}}$$

Exemplo 3.5.2. Seja C o código linear binário com a seguinte matriz de paridade

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

Qual a distância mínima de C?

Designemos por c_i a coluna i de H. A matriz H tem três linhas, ou seja, cada coluna é um vetor em $\mathbb{F}_2^3$. Assim, quaisquer 4 colunas são linearmente dependentes, portanto, $d(C) \leq 3$. Por outro lado

(a) Como não há colunas nulas, qualquer coluna é linearmente independente;

(b) Como não há colunas repetidas, isto é, como $c_i \neq c_j$ se $i \neq j$, então $c_i + c_j \neq \vec{0}$ para $i \neq j$, onde c_i é a coluna i de H, e quaisquer duas colunas são linearmente independentes;

(c) Como $c_1 + c_2 + c_3 = \vec{0}$ (ou $c_2 + c_4 + c_5 = \vec{0}$), há três colunas linearmente dependentes.

Logo, pelo Teorema 3.5.1, $d(C) = 3$.

4 CÓDIGOS CÍCLICOS

Os códigos cíclicos são uma subclasse dos códigos lineares. São códigos de fácil implementação, pois requerem menos informações para a obtenção das palavras de código e têm bons algoritmos de codificação e de decodificação. São exemplos de códigos cíclicos os códigos de Hamming binários, os códigos de Golay G_{11} e G_{23}, os códigos BCH, Reed-Solomon e Goppa, sendo que neste texto desenvolveremos apenas o Código Reed-Solomon.

Definição 4.0.1. Um código C diz-se **cíclico** se:

(i) C é linear (portanto C é subespaço de algum $\mathbb{F}_q^n$) e

(ii) se $x = (x_1, x_2, \ldots, x_{n-1}, x_n) \in C$, então $(x_n, x_1, x_2, \ldots, x_{n-1}) \in C$.

O vetor $(x_n, x_1, x_2, \ldots, x_{n-1}) \in \mathbb{F}_q^n$ diz-se um **desvio cíclico** (**shift**, em inglês) de $x \in \mathbb{F}_q^n$, e iremos denotá-lo por $\sigma(x)$. Portanto, um código é cíclico se é linear e se contém os desvios cíclicos de todas as palavras de código. Assim, se C é um código cíclico , então $\sigma^i(c) \in C$ para todo $c \in C$ e todo $i \in \mathbb{Z}$.

Exemplo 4.0.2.

- Os códigos triviais $\overrightarrow{0}$ e $\mathbb{F}_q^n$ são cíclicos.
- $C_1 = \{0000, 1010, 0101, 1111\} \subset \mathbb{F}_2^4$ é um código cíclico (veja exemplo 4.1.9).
- $C_2 = \{0000, 1001, 0110, 1111\} \subset \mathbb{F}_2^4$ não é cíclico, pois não contém todos os desvios, embora seja linear, gerado pelos vetores 1001 e 0110.
- $C_3 = \{0000, 1010, 0101\} \subset \mathbb{F}_2^4$ não é cíclico porque não é linear, pois falta o vetor $1111 = 1010 + 0101$, mas contém todos os desvios cíclicos.

4.1 POLINÔMIO GERADOR

Apresentaremos algumas noções de álgebra que utilizaremos em seguida. Iremos considerar os anéis de polinômios $\mathbb{F}_q[t]$ e quocientes deste, por isso vamos assumir que R é um anel comutativo com unidade.

Definição 4.1.1. O subconjunto não vazio $I \subset R$ diz-se um **ideal** de R se é fechado para a soma e para o produto por qualquer elemento de R, mais precisamente, se $a + b \in I$ e $ar \in I$ para todo o $a, b \in I$ e $r \in R$.

Dado $a \in R$, o conjunto dos múltiplos de a, $(a) := \{ar : r \in R\}$, é um ideal, chamado de **ideal gerado por a**. Esse gerador a pode não ser único.

Em geral, o conjunto das R-combinações lineares de $a_1, \ldots, a_N$, ou seja, $(a_1, \ldots, a_N) := \{\sum_i a_i r_i : r_i \in R\}$, é um ideal, e $\{a_1, \ldots, a_N\}$ é um conjunto gerador.

Definição 4.1.2. Um ideal $I \in R$ diz-se um **ideal principal** se $I = (a)$ para algum $a \in R$. Se todos os ideais são principais, R diz-se um **anel de ideais principais**.

Exemplo 4.1.3. No anel dos inteiros $\mathbb{Z}$:

- O conjunto dos números pares é um ideal e também é principal: $(2) = \{2x : x \in \mathbb{Z}\}$. O inteiro -2 também é um gerador deste ideal.
- O conjunto dos múltiplo de três é um ideal principal: $(3) = \{3x : x \in \mathbb{Z}\}$, que também pode ser representado por $(15, 12) = \{15x + 12y : x, y \in \mathbb{Z}\}$.

Vamos trabalhar no anel dos polinômios $\mathbb{F}_q[t]$ e o próximo teorema mostra que os ideais neste anel são todos principais.

Teorema 4.1.4. *$\mathbb{F}_q[t]$ é um anel de ideais principais. Assim, se $I \neq \{0\}$ é um ideal, então $I = (g(t))$, onde $g(t)$ é um polinômio mônico de grau mínimo em I. Além disso, este $g(t)$ é único.*

Demonstração. Seja $I \neq \{0\}$ um ideal de $\mathbb{F}_q[t]$. Seja $g(t)$ um polinômio não nulo de grau mínimo em I. Sem perda de generalidade, podemos assumir que $g(t)$ é **mônico** (caso não seja, multiplicamos $g(t)$ pelo inverso do coeficiente do termo líder). O objetivo é ver se este $g(t)$ é um gerador do ideal I.

Seja $a(t)$ um elemento em I qualquer. Pelo algoritmo da divisão de $\mathbb{F}_q[t]$, existem polinômios $q(t)$ e $g(t)$ tais que $a(t) = g(t)q(t) + r(t)$, com $\text{grau}(r(t)) < \text{grau}(g(t))$ ou $r(t) = 0$, portanto $r(t) = a(t) - g(t)q(t)$, que pertence a I, pois $a(t) \in I$ e $g(t) \in I$. Como $g(t)$ tem grau mínimo entre os polinômios não nulos em I, então o resto $r(t)$ é nulo e $a(t) = g(t)q(t) \in (g(t))$. Como $a(t) \in I$ é arbitrário, conclui-se que $I \subset (g(t))$. E como $g(t) \in I$, também se verifica a inclusão inversa $I \supset (g(t))$, assim $I = (g(t))$. □

Precisaremos também do seguinte lema:

Lema 4.1.5. *Seja $g(t) \in \mathbb{F}_q[t]$ tal que $g(t) \mid t^n - 1$. Então $a(t) \equiv g(t)x(t) \pmod{t^n - 1}$ para algum $x(t) \in \mathbb{F}_q[t]$ se e somente se $g(t)$ divide $a(t)$ em $\mathbb{F}_q[t]$.*

Demonstração. Seja $h(t) \in \mathbb{F}_q[t]$ tal que $g(t)h(t) = t^n - 1$. Então:

$$\begin{aligned} & a(t) \equiv g(t)x(t) \pmod{t^n - 1} \\ \implies & a(t) = g(t)x(t) + (t^n - 1)y(t) \text{ para algum } y(t) \in \mathbb{F}_q[t] \\ \implies & a(t) = g(t)x(t) + g(t)h(t)y(t) = g(t)(x(t) + h(t)y(t)) \end{aligned}$$

ou seja, $g(t)$ divide $a(t)$.

Se $g(t) \mid a(t)$

$$\begin{aligned} a(t) = g(t)x(t) &\Longrightarrow a(t) - g(t)x(t) \equiv 0 \pmod{t^n - 1} \\ &\Longrightarrow a(t) \equiv g(t)x(t) \pmod{t^n - 1} \end{aligned}$$

□

Vamos ver algebricamente a condição combinatória dos desvios cíclicos da Definição 4.0.1.

Considere o anel quociente

$$\boxed{R_n = \frac{\mathbb{F}_q[t]}{(t^n - 1)}}$$

Este anel tem uma estrutura natural de espaço vetorial sobre $\mathbb{F}_q$. Considere a aplicação linear sobre $\mathbb{F}_q$

$$\begin{aligned} \varphi \colon \mathbb{F}_q^n &\longrightarrow R_n \\ a = (a_0, a_1, \ldots, a_{n-1}) &\longmapsto a(t) = a_0 + a_1 t + \cdots + a_{n-2}t^{n-2} + a_{n-1}t^{n-1} \end{aligned}$$

Como cada classe de R_n tem um único representante em $\mathbb{F}_q[t]$ de grau menor ou igual a $n-1$ (a saber, o resto da divisão por $t^n - 1$), a aplicação φ é um isomorfismo de espaços vetoriais, além disso:

$$\begin{aligned} \varphi(\sigma(a)) &= \varphi\left(a_{n-1}, a_0, a_1, \ldots, a_{n-2}\right) \\ &= a_{n-1} + a_0 t + \cdots + a_{n-2}t^{n-1} = ta(t) \end{aligned}$$

onde se usou $t^n = 1$, em R_n, no último passo. Portanto:

$$\boxed{\varphi(\sigma(a)) = t\varphi(a)}$$

Ou seja, tomar o shift $\sigma(a)$ em $\mathbb{F}_q^n$ corresponde à multiplicação por t em R_n.

Um jeito de criar códigos cíclicos é tomar um subconjunto $C \subset \mathbb{F}_q^n$ não vazio, de modo que $I = \varphi(C)$ seja um ideal de R_n.

A seguir vamos relacionar os ideais de $\mathbb{F}_q[t]$ com os ideais em R_n.

Considere a aplicação quociente, que é um homomorfismo de anéis (função que preserva produto, soma, 1)

$$\pi \colon \mathbb{F}_q[t] \longrightarrow R_n = \frac{\mathbb{F}_q[t]}{(t^n - 1)}$$

Lema 4.1.6. *Se J é um ideal em $\mathbb{F}_q[t]$ então $\pi(J)$ é um ideal em R_n. Se I é um ideal em R_n, então $\pi^{-1}(I)$ é um ideal em $\mathbb{F}_q[t]$ que contém $t^n - 1$.*

Demonstração. O resultado segue da definição de ideal, tendo em conta que a imagem e a pré-imagem de conjuntos são, respectivamente:

$$\pi(J) := \{\pi(j) \in R_n : j \in J\}$$

e

$$\pi^{-1}(I) := \{i \in \mathbb{F}_q[t] : \pi(i) = [i] \in I\}$$

A título de exemplo vamos demonstrar que $\pi^{-1}(I)$ é um ideal de $\mathbb{F}_q[t]$.

Dado $\pi^{-1}(I) \subseteq \mathbb{F}[t]$, tome $f, g \in \pi^{-1}(I) \iff \pi(g) \in I$ e $\pi(f) \in I$. Como o ideal I é fechado por soma, temos $\pi(g) + \pi(f) \in I \implies \pi(g + f) \in I \implies g + f \in \pi^{-1}(I)$.

Dados $\lambda \in \mathbb{F}_q[t]$, $f \in \pi^{-1}(I)$, devemos mostrar que $\lambda f \in \pi^{-1}(I)$, ou seja, $\pi(\lambda f) \in I$. Como $\pi(\lambda f) = \pi(\lambda) \cdot \pi(f)$ e $\pi(\lambda) \in R_n$ e $\pi(f) \in I$ então $\pi(\lambda) \cdot \pi(f) \in I$. □

Pode-se verificar que, como consequência deste lema, a aplicação π define uma correspondência biunívoca entre os ideais no quociente R_n e os ideais contendo $t^n - 1$ no anel de polinômios $\mathbb{F}_q[t]$. Como consequência do lema acima, temos

Teorema 4.1.7. *R_n é um anel de ideais principais. Assim, se $I \neq \{0\}$ é um ideal em R_n, então $I = (g(t))$, onde $g(t)$ é um polinômio mônico de grau mínimo em I. Além disso, este $g(t)$ é único, $g(t) \mid t^n - 1$, e assim $t \nmid t^n - 1 \implies g_0 = g(0) \neq 0$.*

Definição 4.1.8. Dado um ideal $I \subset R_n$, tal que $\pi^{-1}(I) = (g(t))$, o polinômio $g(t) \in \mathbb{F}_q[t]$ é chamado **polinômio gerador** do código cíclico $C = \varphi^{-1}(I)$.

Exemplo 4.1.9. Considere

$$\pi \colon \mathbb{F}_2[t] \longrightarrow R_4 = \mathbb{F}_2[t]/(t^4 - 1)$$

Onde $R_4 = \dfrac{\mathbb{F}_2[t]}{(t^4 - 1)} = \{a_0 + a_1 t + a_2 t^2 + a_3 t^3 \mid a_i \in \mathbb{F}_2\}$, cujos elementos são dados pelos

possíveis restos na divisão por $t^4 - 1$. Este espaço vetorial possui 2^4 elementos:

$$
\begin{aligned}
0 + 0t + 0t^2 + 0t^3 &= (0000) \\
1 + 0t + 0t^2 + 0t^3 &= (1000) \\
0 + 1t + 0t^2 + 0t^3 &= (0100) \\
0 + 0t + 1t^2 + 0t^3 &= (0010) \\
0 + 0t + 0t^2 + 1t^3 &= (0001) \\
1 + 1t + 0t^2 + 0t^3 &= (1100) \\
1 + 0t + 1t^2 + 0t^3 &= (1010) \\
1 + 0t + 0t^2 + 1t^3 &= (1001) \\
0 + 1t + 1t^2 + 0t^3 &= (0110) \\
0 + 1t + 0t^2 + 1t^3 &= (0101) \\
0 + 0t + 1t^2 + 1t^3 &= (0011) \\
1 + 1t + 1t^2 + 0t^3 &= (1110) \\
1 + 1t + 0t^2 + 1t^3 &= (1101) \\
1 + 0t + 1t^2 + 1t^3 &= (1011) \\
0 + 1t + 1t^2 + 1t^3 &= (0111) \\
1 + 1t + 1t^2 + 1t^3 &= (1111)
\end{aligned}
$$

Agora, considere o ideal $I = \{0, 1 + t^2, t + t^3, 1 + t + t^2 + t^3\} \subset R_4$. De acordo com o Lema 4.1.6, $\pi^{-1}(I) = (1 + t^2, t + t^3, 1 + t + t^2 + t^3, t^4 - 1)$, mas como $t + t^3 = t(1 + t^2)$, $1 + t + t^2 + t^3 = (1 + t^2)(t + 1)$ e $t^4 - 1 = (t^2 + 1)(t^2 - 1)$, ou seja, todos estes polinômios são múltiplos de $(1 + t^2)$, então $\pi^{-1}(I) = (1 + t^2) \subset \mathbb{F}_2[t]$. Logo, o polinômio gerador do código cíclico $\varphi(I) = \{0000, 1010, 0101, 1111\}$ é $g(t) = t^2 + 1 \in \mathbb{F}_2[t]$.

A seguir, veremos como encontrar a matriz geradora e a matriz paridade dos códigos cíclicos.

4.2 MATRIZ GERADORA E MATRIZ DE PARIDADE

Teorema 4.2.1. *Seja $g(t) = g_0 + g_1t + g_2t^2 + \cdots + g_rt^r$ o polinômio gerador do código cíclico $C \subset R_n$. Então*

$$G_{n\times(n-r)} = \begin{bmatrix} g_0 & 0 & \cdots & 0 & 0 \\ g_1 & g_0 & \ddots & \vdots & \vdots \\ \vdots & g_1 & \ddots & 0 & \vdots \\ g_r & \vdots & \ddots & g_0 & 0 \\ 0 & g_r & & g_1 & g_0 \\ \vdots & 0 & \ddots & \vdots & g_1 \\ \vdots & \vdots & \ddots & g_r & \vdots \\ 0 & 0 & \cdots & 0 & g_r \end{bmatrix} = \begin{bmatrix} | & | & & | \\ g(t) & tg(t)\dots & t^{n-r-1}g(t) \\ | & | & & | \end{bmatrix}$$

é uma matriz geradora de C *e* $\dim C = n - r = n - \text{grau}(g(t))$*, ou seja, o grau de* $g(t)$ *é a redundância do código* C*.*

Demonstração. Queremos provar que as colunas da matriz geradora são linearmente independentes. Sejam $v_1, v_2, \dots, v_{n-r}$ os vetores coluna da matriz geradora e $\lambda_1, \lambda_2, \dots, \lambda_{n-r}$ escalares. Os vetores $v_1, v_2, \dots, v_{n-r}$ são linearmente independentes se

$$\lambda_1 v_1 + \lambda_2 v_2 + \cdots + \lambda_{n-r} v_{n-r} = 0$$

$$\implies \lambda_1 = \lambda_2 = \cdots = \lambda_{n-r} = 0.$$

Olhando para as primeiras $r + 2$ coordenadas, teremos o seguinte sistema linear

$$\begin{aligned} \lambda_1 g_0 &= 0 \\ \lambda_1 g_1 + \lambda_2 g_0 &= 0 \\ \ddots \quad &\vdots \\ \lambda_1 0 + \lambda_2 g_r + \cdots + \lambda_{n-r} g_0 &= 0 \end{aligned}$$

mas como $g_0 \neq 0$, pelo Teorema 4.1.7, então

$$\lambda_1 g_0 = 0 \implies \lambda_1 = 0$$

$$\lambda_1 g_1 + \lambda_2 g_0 = 0 \implies \lambda_2 = 0$$

$$\vdots$$

$$\lambda_1 0 + \lambda_2 g_r + \cdots + \lambda_{n-r} g_0 = 0 \implies \lambda_{n-r} = 0$$

Logo, as colunas da matriz geradora são linearmente independentes. Veremos que as colunas também formam um conjunto gerador, como espaço vetorial, do código C. Seja $a(t) \in C = (g(t))$. Então $\text{grau}(a(t)) < n$ e, pelo Lema 4.1.5, $a(t) = g(t)x(t)$ para algum polinômio $x(t) \in \mathbb{F}_q[t]$. Como $\text{grau}(a(t)) = \text{grau}(g(t)x(t)) = \text{grau}(g(t)) + \text{grau}(x(t))$, então

grau$(x(t)) < n - r$, ou seja,

$$x(t) = x_0 + x_1 t + \cdots + x_{n-r-1} t^{n-r-1}$$

e

$$a(t) = g(t)x(t) = x_0 g(t) + x_1 t g(t) + \cdots + x_{n-r-1} t^{n-r-1} g(t),$$

ou seja, $a(t)$ é combinação linear de $g(t), tg(t), \ldots, t^{n-r-1}g(t)$, que são precisamente as colunas da matriz G. Uma vez que as $n - r$ colunas de G formam uma base de C, conclui-se que C tem dimensão $n - r$. □

Definição 4.2.2. Se C é um código cíclico, de comprimento n, com polinômio gerador $g(t)$, então $h(t) = \dfrac{t^n - 1}{g(t)} \in \mathbb{F}_q[t]$ diz-se o **polinômio de paridade** de C.

Uma vez que $g(t)$ é mônico, então $h(t)$ também é, pois $h(t)g(t) = t^n - 1$.

Teorema 4.2.3. *Seja C um código cíclico, de comprimento n e dimensão k, com polinômio de paridade $h(t) = h_0 + h_1 t + \cdots + h_k t^k$. Então a matriz*

$$H_{(n-k)\times n} = \begin{bmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \vdots & \\ 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & h_0 & 0 \\ 0 & \cdots & \cdots & 0 & h_k & h_{k-1} & \cdots & h_0 \end{bmatrix}$$

é a matriz de paridade para C.

Demonstração. Como $h(t)$ é um polinômio mônico, temos que $h_k = 1 \neq 0$. Portanto, H é a matriz de uma transformação sobrejetora $H\colon \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$, pois sua imagem contém as primeiras $n - k$ colunas, que são linearmente independentes já que $h_k \neq 0$ (veja a demonstração de independência linear das colunas de G no teorema anterior, em que obtemos um sistema triangular). Logo pelo teorema do núcleo e da imagem temos que a $\dim \ker H = k$, que é igual a $\dim C = n - r = k$, pelo Teorema 4.2.1. Assim, para mostrar que H é uma matriz paridade para C, basta ver que $Hc = 0$ se $c \in C$.

Temos $h(t)g(t) = t^n - 1$. Ilustraremos a ideia da prova para um caso menor, por exemplo

$$\begin{gathered}(h_0 + h_1 t + h_2 t^2 + h_3 t^3)(g_0 + g_1 t + g_2 t^2) = t^5 - 1 \\ \iff h_0 g_0 + (h_0 g_1 + h_1 g_0)t + (h_0 g_2 + h_1 g_1 + h_2 g_0)t^2 \\ +(h_1 g_2 + h_2 g_1 + h_3 g_0)t^3 + (h_2 g_2 + h_3 g_1)t^4 + h_3 g_2 t^5 = t^5 - 1\end{gathered}$$

Comparando os coeficientes, obtemos

$$\begin{aligned} h_2g_0 + h_1g_1 + h_0g_2 = 0 &\longrightarrow \text{Termos de grau 2} \\ h_3g_0 + h_2g_1 + h_1g_2 = 0 &\longrightarrow \text{Termos de grau 3} \\ h_3g_1 + h_2g_2 = 0 &\longrightarrow \text{Termos de grau 4} \end{aligned}$$

Assim, em notação matricial,

$$\begin{bmatrix} h_3 & h_2 & h_1 & h_0 & 0 \\ 0 & h_3 & h_2 & h_1 & h_0 \end{bmatrix} \cdot \begin{bmatrix} g_0 & 0 & 0 \\ g_1 & g_0 & 0 \\ g_2 & g_1 & g_0 \\ 0 & g_2 & g_1 \\ 0 & 0 & g_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

No caso geral, $g(t)h(t) = t^n - 1$ e comparando coeficientes, temos, para $1 \leq i \leq n-1$,

$$g_0h_i + g_1h_{i-1} + \cdots + g_ih_0 = 0$$

Logo, em notação matricial, obtemos $HG = 0$. □

Exemplo 4.2.4. Seja $g(t) = 1 + 8t + 5t^2 + 3t^3 + t^4 \in \mathbb{F}_{11}[t]$ o polinômio gerador do código cíclico em $R_{10} = \mathbb{F}_{11}[t]/(t^{10} - 1)$. Utilizaremos o teorema acima para determinar a matriz geradora G. A redundância do código cíclico gerado por este polinômio é igual a 4, pois o grau$(g(t)) = 4$, enquanto a $\dim(C) = n - \text{grau}(g(t)) = 10 - 4 = 6$. Desta forma, temos que a matriz geradora é

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 0 & 0 & 0 & 0 \\ 5 & 8 & 1 & 0 & 0 & 0 \\ 3 & 5 & 8 & 1 & 0 & 0 \\ 1 & 3 & 5 & 8 & 1 & 0 \\ 0 & 1 & 3 & 5 & 8 & 1 \\ 0 & 0 & 1 & 3 & 5 & 8 \\ 0 & 0 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Vamos encontrar a matriz paridade.

$$h(t) = \frac{t^{10} - 1}{1 + 8t + 5t^2 + 3t^3 + t^4} = 10 + 8t + 7t^2 + 6t^3 + 4t^4 + 8t^5 + t^6 \in \mathbb{F}_{11}[t]$$

$$H = \begin{bmatrix} 1 & 8 & 4 & 6 & 7 & 8 & 10 & 0 & 0 & 0 \\ 0 & 1 & 8 & 4 & 6 & 7 & 8 & 10 & 0 & 0 \\ 0 & 0 & 1 & 8 & 4 & 6 & 7 & 8 & 10 & 0 \\ 0 & 0 & 0 & 1 & 8 & 4 & 6 & 7 & 8 & 10 \end{bmatrix}$$

Note que $HG = 0$.

Observação 4.2.5. *As divisões de polinômios foram feitas com auxílio do programa PARI/GP.*

5 CÓDIGOS DE REED-SOLOMON (RS)

Os códigos de Reed-Solomon são códigos cíclicos que foram introduzidos por Irving S. Reed e Gustave Solomon em 1960. A vantagem deste código é que com uma boa escolha do polinômio gerador é possível controlar a distância mínima e desta forma corrigir erros acumulados. Eles são muito utilizados em sistemas de transmissão de dados como comunicação via satélite, comunicações de missões espaciais, aDSL, WiMAX (similar ao Wi-Fi), DVB (Transmissão de Vídeo Digital), sistemas RAID 6 (sistema de armanezamento) e sistemas de telecomunicações como DWDM (tecnologia que usa múltiplos lasers para transmitir muitos comprimentos de onda de luz simultaneamente), tecnologias de consumo como CDs, DVDs, QRCode.

5.1 ORDEM E ELEMENTO PRIMITIVO

Para encontrarmos o polinômio gerador de um código de Reed-Solomon, precisamos de um elemento primitivo, então vamos ver a definição:

Definição 5.1.1. Seja $\alpha \in \mathbb{F}_q^{\times}$.

(i) A **ordem** $\text{ord}(\alpha)$ de α é o menor inteiro $d > 0$ tal que $\alpha^d = 1$.

(ii) Dizemos que α é um **elemento primitivo** de $\mathbb{F}_q$ se, e somente se, $\{\alpha^t, t \in \mathbb{N}\} = \mathbb{F}_q^{\times}$.

Teorema 5.1.2. *Seja $\alpha \in \mathbb{F}_q^{\times}$.*

(i) Se $d \in \mathbb{N}$ é tal que $\alpha^d = 1$ então $\text{ord}(\alpha) \mid d$.

(ii) $\alpha^{q-1} = 1$.

(iii) $\text{ord}(\alpha)$ é um divisor de $q - 1$.

(iv) O elemento α é primitivo se, e só se, $\text{ord}(\alpha) = q - 1$.

Demonstração. (i) Fixemos $\alpha \in \mathbb{F}_q^{\times}$ e seja d um número inteiro tal que $\alpha^d = 1$. Sejam r e s inteiros tais que $d = \text{ord}(\alpha)s + r$ e $0 \leq r < \text{ord}(\alpha)$ (pelo algoritmo da divisão). Então

$$1 = \alpha^d = \alpha^{\text{ord}(\alpha)s+r} = (\alpha^{\text{ord}(\alpha)})^s \alpha^r = \alpha^r$$

Portanto $r = 0$, por definição de $\text{ord}(\alpha)$, ou seja, $\text{ord}(\alpha)$ divide d.

(ii) Seja $\alpha \neq 0$ e $\mathbb{F}_q^\times = \{b_1, \cdots, b_{q-1}\}$. Como $\alpha \neq 0$, também temos que $\mathbb{F}_q^\times = \{\alpha b_1, \cdots, \alpha b_{q-1}\}$. Assim, multiplicando todos os elementos de $\mathbb{F}_q^\times$, temos

$$b_1 \cdots b_{q-1} = (\alpha b_1) \cdots (\alpha b_{q-1}) \Longrightarrow b_1 \cdots b_{q-1} = (\alpha^{q-1})(b_1 \cdots b_{q-1})$$

Logo, obtém-se $\alpha^{q-1} = 1$.

Pelos itens (i) e (ii), concluímos que $\text{ord}(\alpha)$ divide $q-1$.

(iii) Para todo $\alpha \in \mathbb{F}_q^\times$ temos $\{\alpha^t, t \in \mathbb{N}\} \subset \mathbb{F}_q^\times$. Note que

$$\{\alpha^t, t \in \mathbb{N}\} = \{1, \alpha, \alpha^2, \cdots, \alpha^{\text{ord}\,\alpha - 1}\}$$

é um conjunto com $\text{ord}(\alpha)$ elementos. De fato, para qualquer $t \in \mathbb{N}$ temos $\alpha^t = \alpha^r$ onde r é o resto na divisão de t por $\text{ord}(\alpha)$; por outro lado, os elementos $1, \alpha, \alpha^2, \cdots, \alpha^{\text{ord}\,\alpha - 1}$ são distintos pois caso $\alpha^i = \alpha^j$ com $0 \leq i < j < \text{ord}(\alpha)$, então $\alpha^{j-i} = 1$ com $0 < j - i < \text{ord}(\alpha)$, o que é um absurdo. Assim, α é um elemento primitivo em $\mathbb{F}_q$, isto é, $\{\alpha^t, t \in \mathbb{N}\} = \mathbb{F}_q^\times$, se, e somente se, $\text{ord}(\alpha) = q-1$. □

Exemplo 5.1.3. Vamos encontrar os elementos primitivos de $\mathbb{F}_{11}$. Temos $\mathbb{F}_{11} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, então $\varphi(11) = 10$ (número de elemento que são primos com 11). Para encontrarmos as ordens dos elementos procuramos os divisores de 10, que são $\{1, 2, 5, 10\}$, as possíveis ordens.

Vamos começar procurando $\text{ord}_{11}\, 2$: sabemos que $2^{10} \equiv 1 \pmod{11}$, pelo Pequeno Teorema de Fermat (PTF), mas precisamos testar também para $2^5, 2^2$ e 2^1. Temos $2^5 = 32 \equiv 10 \pmod{11}$, $2^2 \equiv 4 \pmod{11}$ e $2^1 \equiv 2 \pmod{11}$. Logo, $\text{ord}_{11}\, 2 = 10 = \varphi(11)$, então 2 é elemento primitivo de $\mathbb{F}_{11}$.

Da mesma forma, para $\text{ord}_{11}\, 3$ temos que $3^{10} \equiv 1 \pmod{11}$, pelo PTF; $3^5 = 243 \equiv 1 \pmod{11}$; $3^2 \equiv 9 \pmod{11}$; $3^1 \equiv 3 \pmod{11}$. Logo, $\text{ord}_{11}\, 3 = 5 \neq \varphi(11)$, assim, 3 não é elemento primitivo de $\mathbb{F}_{11}$.

Continuando desta forma, obtemos a seguinte tabela:

Elemento de $\mathbb{F}_{11}^{\times}$	**Ordem**	**Primitivo?**
1	1	não
2	10	sim
3	5	não
4	5	não
5	5	não
6	10	sim
7	10	sim
8	10	sim
9	5	não
10	2	não

Desta forma, encontramos todos os elementos primitivos de $\mathbb{F}_{11}$ que são: 2, 6, 7 e 8.

Exemplo 5.1.4. Encontrar um elemento primitivo de $\mathbb{F}_9 = \dfrac{\mathbb{F}_3[x]}{(x^2+1)}$.

Temos que x^2+1 é um polinômio irredutível em $\mathbb{F}_3[x]$, pois não possui raízes em $\mathbb{F}_3$. Assim,

$$\begin{aligned}\frac{\mathbb{F}_3[x]}{(x^2+1)} &= \{a_0 + a_1 t \mid a_i \in \mathbb{F}_3 \text{ e } t^2 = -1 = 2\} \\ &= \{0, 1, 2, t, 2t, 1+t, 1+2t, 2+t, 2+2t\},\end{aligned}$$

que são os possíveis restos na divisão por x^2+1. Para encontrarmos as ordens dos elementos procuramos os divisores de 8, ou seja, 8, 4, 2, 1, que são as possíveis ordens.

Ordem de t: $t^2 = 2$; $t^4 = t^2 \cdot t^2 = 2 \cdot 2 = 4 = 1$.

Desta forma, a $\text{ord}(t) = 4$. Assim, t não é elemento primitivo de $\mathbb{F}_9$.

Ordem de $(1+t)$:

$$\begin{aligned}(1+t)^2 &= 1 + 2t + t^2 = 1 + 2t + 2 = 3 + 2t = 0 + 2t = 2t; \\ (1+t)^4 &= (1+t)^2(1+t)^2 = 2t \cdot 2t = 4t^2 = 4 \cdot 2 = 8 = 2; \\ (1+t)^8 &= (1+t)^4(1+t)^4 = 2 \cdot 2 = 4 = 1\end{aligned}$$

Desta forma, $\text{ord}(1+t) = 8$. Logo $1+t$ é elemento primitivo de $\mathbb{F}_9$.

Continuando desta forma, obtemos a seguinte tabela:

Elemento de $\mathbb{F}_9^\times$	Ordem	Primitivo?
1	1	não
2	2	não
t	4	não
$2t$	4	não
$1+t$	8	sim
$2+t$	8	sim
$1+2t$	8	sim
$2+2t$	8	sim

Assim, os elementos primitivos de $\mathbb{F}_9$ são: $1+t, 2+t, 1+2t$ e $2+2t$.

5.2 CÓDIGOS DE REED-SOLOMON

Definição 5.2.1. Um **código Reed-Solomon** q-ário é um código cíclico de comprimento $q-1$, com polinômio gerador

$$g(t) = (t-\alpha^1)(t-\alpha^2)\cdots(t-\alpha^{\delta-1})$$

com $2 \leq \delta \leq q-1$, onde α é um elemento primitivo de $\mathbb{F}_q$.

Observação 5.2.2. *(i) Se α é um elemento primitivo de $\mathbb{F}_q$, o polinômio $t^{q-1}-1 \in \mathbb{F}_q[t]$ tem a seguinte fatoração*

$$t^{q-1}-1 = (t-1)(t-\alpha)(t-\alpha^2)\cdots(t-\alpha^{q-2}).$$

Pois $t^{q-1}-1$ tem $1, \alpha, \alpha^2, \cdots, \alpha^{q-2}$ como raízes pelo teorema 5.1.2 item (ii).

Então, o polinômio $g(t)$ é um divisor de $t^{q-1}-1$ e $g(t)$ é o polinômio gerador de um código cíclico e suas raízes são todas distintas.

(ii) Como o $\text{grau}(g(t)) = \delta - 1$, *a dimensão de C é* $\dim(C) = (q-1)-(\delta-1) = q-\delta$, *pelo Teorema 4.2.1.*

(iii) Não existem códigos Reed-Solomon binários, pois $2 \leq \delta \leq q-1 \Longrightarrow q \geq 3$.

Exemplo 5.2.3. Como 2 é um elemento primitivo de $\mathbb{F}_{11}$, vamos escolher o seguinte polinômio gerador:

$$g(t) = (t-2)(t-2^2)(t-2^3)(t-2^4) = 1+8t+5t^2+3t^3+t^4$$

o qual é o polinômio gerador de um código Reed-Solomon de parâmetros (10, 6). Note que este polinômio foi utilizado no Exemplo 4.2.4.

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 0 & 0 & 0 & 0 \\ 5 & 8 & 1 & 0 & 0 & 0 \\ 3 & 5 & 8 & 1 & 0 & 0 \\ 1 & 3 & 5 & 8 & 1 & 0 \\ 0 & 1 & 3 & 5 & 8 & 1 \\ 0 & 0 & 1 & 3 & 5 & 8 \\ 0 & 0 & 0 & 1 & 3 & 5 \\ 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

e o polinômio de paridade é o mesmo do Exemplo 4.2.4.

Como $d(C) \geq \delta$, pelo Teorema 5.3.2, o qual veremos em seguida, podemos determinar que a distância mínima deste código é $d(C) \geq 5$, pois $\delta - 1 = 4$, temos que $\delta = 5$, neste caso poderemos corrigir até dois erros de transmissão.

5.3 DISTÂNCIA MÍNIMA

Observe que os polinômios $c(t)$ em R_n, múltiplos de $g(t)$, são aqueles que se anulam nas raízes de $g(t)$. Logo,

Proposição 5.3.1. *Seja C um código Reed-Solomon q-ário, com polinômio gerador $g(t) = (t-\alpha^1)(t-\alpha^2)\cdots(t-\alpha^{\delta-1})$. Então*

$$C = \{c(t) \in R_{q-1} : c(\alpha^i) = 0, \forall i = 1, \ldots, \delta - 1\}.$$

Teorema 5.3.2. *Seja C um código Reed-Solomon de parâmetros $(q-1, q-\delta)_q$. Então $d(C) \geq \delta$.*

Demonstração. Seja $g(t) = (t-\alpha^1)(t-\alpha^2)\cdots(t-\alpha^{\delta-1})$ o polinômio gerador de C. Suponhamos, por absurdo, que $d(C) = d < \delta$ e seja $c(t) = c_0 + c_1 t + \cdots + c_{n-1}t^{n-1} \in C$ com peso $w(c(t)) = d$. Seja $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ o vetor correspondente a $c(t)$. Pela Proposição 5.3.1, $c(\alpha^i) = 0$ para qualquer $i = 1, \ldots, \delta - 1$. Por outro lado,

$$c(\alpha^i) = c_0 + c_1\alpha^i + \cdots + c_{n-1}(\alpha^i)^{n-1} = (1, \alpha^i, (\alpha^i)^2, \ldots, (\alpha^i)^{n-1}) \cdot c.$$

Portanto $Ac = 0$, onde A é a matriz de Vandermonde

$$A = \begin{bmatrix} 1 & \alpha^1 & (\alpha^1)^2 & \cdots & (\alpha^1)^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \cdots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{\delta-1} & (\alpha^{\delta-1})^2 & \cdots & (\alpha^{\delta-1})^{n-1} \end{bmatrix}$$

uma matriz com $\delta - 1$ linhas e n colunas. Sejam $i_1, \ldots, i_d$ os índices tais que $c_{i_j} \neq 0$. Seja A' a matriz formada pelas colunas $i_1 + 1, i_2 + 1, \ldots, i_d + 1$ de A. Então A' tem $\delta - 1$ linhas e d colunas. Como $d \leq \delta - 1$, a matriz A'' formada pelas d primeiras linhas de A' é uma matriz quadrada $d \times d$ e

$$A'' \begin{bmatrix} c_{i_1} \\ \vdots \\ c_{i_d} \end{bmatrix} = 0$$

portanto $\det(A'') = 0$, pois $(c_{i_1}, \ldots, c_{i_d})$ é uma solução não nula do sistema linear homogêneo $A''x = 0$. Por outro lado, como

$$A'' = \begin{bmatrix} (\alpha^1)^{i_1} & (\alpha^1)^{i_2} & \ldots & (\alpha^1)^{i_d} \\ (\alpha^2)^{i_1} & (\alpha^2)^{i_2} & \cdots & (\alpha^2)^{i_d} \\ \vdots & \vdots & & \vdots \\ (\alpha^d)^{i_1} & (\alpha^d)^{i_2} & \ldots & (\alpha^d)^{i_d} \end{bmatrix}$$

usando as propriedades de multilinearidade do determinante nas colunas obtém-se

$$\det(A'') = \prod_{j=1}^{d} \alpha^{i_j} \det \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \cdots & \alpha^{i_d} \\ \vdots & \vdots & & \vdots \\ (\alpha^{d-1})^{i_1} & (\alpha^{d-1})^{i_2} & \cdots & (\alpha^{d-1})^{i_d} \end{bmatrix}$$

$$= \prod_{j=1}^{d} \alpha^{i_j} \prod_{1 \leq k < l \leq d} (\alpha^{i_l} - \alpha^{i_k})$$

Logo, $\det(A'') \neq 0$, pois $\alpha^{i_l} \neq \alpha^{i_k}$, porque α é um elemento primitivo e $d \leq \delta - 1 \leq q - 2$. Como não podemos ter simultaneamente $\det(A'') = 0$ e $\det(A'') \neq 0$, concluímos que $d \geq \delta$. □

5.4 ALGORITMO DE PETERSON-GORENSTEIN-ZIERLER (PGZ)

O código cíclico também é um código linear, o qual já vimos o algoritmo de codificação e detecção de erro. Nesta seção veremos o algoritmo de Peterson-Gorenstein-Zierler

para correção de erros do Código de Reed-Solomon.

5.4.1 Descrição do algoritmo

A mensagem transmitida é vista como os coeficientes de um polinômio $p(t)$ em $\mathbb{F}_q[t]$, que é divisível por um polinômio gerador $g(t)$. Sejam

$$p(t) = \sum_{i=0}^{q-2} c_i t^i$$

$$g(t) = \prod_{j=1}^{\delta-1} (t - \alpha^j),$$

onde α é um elemento primitivo de $\mathbb{F}_q$. Seja

$$e(t) = \sum_{i=0}^{q-2} e_i t^i$$

o erro e $y(t)$ o polinômio recebido, de modo que

$$y(t) = p(t) + e(t)$$

Se há ν erros em potências distintas i_k de t, então

$$e(t) = \sum_{k=1}^{\nu} e_{i_k} t^{i_k}$$

O objetivo do decodificador é encontrar as posições dos erros i_k e os valores de erro nessas posições e_{i_k}. Para isto, definimos

Definição 5.4.1. Seja $\nu \leq \lfloor \frac{\delta-1}{2} \rfloor$, o número de erros a serem corrigidos.

(a) As **síndromes** $S_1, \ldots, S_{\delta-1}$ são definidas como

$$S_j = y(\alpha^j)$$

(b) Os **localizadores** de erro $X_1, \ldots, X_\nu$ são definidos por

$$X_k = \alpha^{i_k}$$

(c) Os **valores de erro** $Y_1, \ldots, Y_\nu$ são definidos como

$$Y_k = e_{i_k}$$

(d) O **polinômio localizador de erros** $\Lambda(x)$ é definido como

$$\Lambda(x) = \prod_{k=1}^{\nu}(1 - xX_k) = 1 + \Lambda_1 x^1 + \Lambda_2 x^2 + \cdots + \Lambda_\nu x^\nu$$

Note que os zeros de $\Lambda(x)$ são os recíprocos X_k^{-1}.

O algoritmo seguirá os seguintes passos:

1. Calcular as síndromes S_j para o polinômio recebido $y(t)$;

2. Determinar o polinômio do localizador de erros $\Lambda(x)$ a partir das síndromes S_j;

3. Calcular as raízes $X_k^{-1} = \alpha^{-i_k}$ do polinômio de localização do erro $\Lambda(x)$, obtendo assim as posições i_k dos erros;

4. Calcular os valores de erro Y_k, logo o polinômio de erro $e(t) = \sum_{1 \leq k \leq \nu} Y_k t^{i_k}$

5. Corrigir os erros, obtendo a mensagem original $p(t) = y(t) - e(t)$.

Antes de vermos o caso geral, vejamos um caso particular com $\delta = 5$ e $\nu = 2$. Note primeiro que como $g(t) \mid p(t)$ e $g(t) = (t-\alpha^1)(t-\alpha^2)(t-\alpha^3)(t-\alpha^4)$, de $y(t) = p(t) + e(t)$ temos que as síndromes S_j são da seguinte forma:

$$\begin{cases} S_1 = y(\alpha) = e(\alpha) \\ S_2 = y(\alpha^2) = e(\alpha^2) \\ S_3 = y(\alpha^3) = e(\alpha^3) \\ S_4 = y(\alpha^4) = e(\alpha^4) \end{cases} \tag{5.1}$$

Note que o polinômio de erro $e(t) = e_0 + e_1 t + \cdots + e_{q-2}t^{q-2}$ tem no máximo $\nu = 2$ coeficientes não nulos. Suponhamos que saibamos as posições do erro, por exemplo $e(t) = e_1 t + e_5 t^5$, vamos mostrar que conhecendo estas posições podemos encontrar o erro $e(t)$. Como

$$\begin{cases} S_1 = y(\alpha) = e(\alpha) = e_1\alpha + e_5\alpha^5 \\ S_2 = y(\alpha^2) = e(\alpha^2) = e_1\alpha^2 + e_5\alpha^{10} \end{cases} \tag{5.2}$$

basta resolver o sistema em e_1 e e_5

$$\begin{bmatrix} \alpha & \alpha^5 \\ \alpha^2 & \alpha^{10} \end{bmatrix} \begin{bmatrix} e_1 \\ e_5 \end{bmatrix} = \begin{bmatrix} S_1 = e(\alpha) \\ S_2 = e(\alpha^2) \end{bmatrix}. \tag{5.3}$$

Para descobrir onde está o erro, definimos $X_k = \alpha^{i_k}$ e $Y_k = e_{i_k}$, neste caso, temos: $X_1 = \alpha$, $X_2 = \alpha^5$, $Y_1 = e_1$ e $Y_2 = e_5$. O polinômio localizador de erro será o seguinte:

$$\Lambda(x) = (1 - xX_1)(1 - xX_2) = 1 - x(X_1 + X_2) + x^2 X_1 X_2 = 1 + \Lambda_1 x^1 + \Lambda_2 x^2$$

com $\Lambda_1 = -(X_1 + X_2)$ e $\Lambda_2 = X_1 X_2$ e as síndromes serão dadas por $S_j = e(\alpha^j) = Y_1 X_1^j + Y_2 X_2^j$, ou seja,

$$S_1 = Y_1 X_1^1 + Y_2 X_2^1$$

$$S_2 = Y_1 X_1^2 + Y_2 X_2^2$$

$$S_3 = Y_1 X_1^3 + Y_2 X_2^3$$

$$S_4 = Y_1 X_1^4 + Y_2 X_2^4$$

Verifiquemos que os coeficientes Λ_i satisfazem

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix} \tag{5.4}$$

Substituindo as expressões das síndromes e dos Λ_i's em $S_1\Lambda_2 + S_2\Lambda_1$, obtemos

$$(Y_1 X_1^1 + Y_2 X_2^1)(X_1 X_2) - (Y_1 X_1^2 + Y_2 X_2^2)(X_1 + X_2) =$$

$$Y_1 X_1^2 X_2 + Y_2 X_1 X_2^2 - Y_1 X_1^3 - Y_1 X_1^2 X_2 - Y_2 X_1 X_2^2 - Y_2 X_2^3 =$$

$$= -Y_1 X_1^3 - Y_2 X_2^3 = -S_3$$

E substituindo em $S_2\Lambda_2 + S_3\Lambda_1$

$$(Y_1 X_1^2 + Y_2 X_2^2)(X_1 X_2) - (Y_1 X_1^3 + Y_2 X_2^3)(X_1 + X_2) =$$

$$= Y_1 X_1^3 X_2 + Y_2 X_1 X_2^3 - Y_1 X_1^4 - Y_1 X_1^3 X_2 - Y_2 X_1 X_2^3 - Y_2 X_2^4 =$$

$$= -Y_1X_1^4 - Y_2X_2^4 = -S_4$$

Resolvendo (5.4) encontraremos $\Lambda(x)$, cujos zeros são os inversos de X_k^{-1}, que nos dão as posições dos erros. Daí substituindo em (5.2), encontraremos os valores dos erros Y_k e, assim, a mensagem enviada.

Teorema 5.4.2. *(i) Os coeficientes Λ_i do polinômio de localização do erro satisfazem o sistema linear*

$$\begin{bmatrix} S_1 & S_2 & \cdots & S_\nu \\ S_2 & S_3 & \cdots & S_{\nu+1} \\ \vdots & \vdots & & \vdots \\ S_\nu & S_{\nu+1} & \cdots & S_{2\nu-1} \end{bmatrix} \begin{bmatrix} \Lambda_\nu \\ \Lambda_{\nu-1} \\ \vdots \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_{\nu+1} \\ -S_{\nu+2} \\ \vdots \\ -S_{\nu+\nu} \end{bmatrix}$$

(ii) Os valores de erro Y_k satisfazem o sistema linear

$$\begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_\nu^1 \\ X_1^2 & X_2^2 & \cdots & X_\nu^2 \\ \vdots & \vdots & & \vdots \\ X_1^{\delta-1} & X_2^{\delta-1} & \cdots & X_\nu^{\delta-1} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_\nu \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{\delta-1} \end{bmatrix}$$

Demonstração. Note inicialmente que, como $p(t)$ é divisível pelo gerador $g(t)$, $p(\alpha^j) = 0$ para $j = 1, 2, \ldots, \delta - 1$, temos

$$\begin{aligned} S_j = y(\alpha^j) &= p(\alpha^j) + e(\alpha^j) \\ &= e(\alpha^j) = \sum_{k=1}^{\nu} e_{i_k} \left(\alpha^j\right)^{i_k} = \sum_{k=1}^{\nu} Y_k X_k^j \end{aligned} \tag{5.5}$$

o que prova (ii). Por outro lado, como X_k^{-1} é raiz de $\Lambda(x)$, temos

$$\Lambda(X_k^{-1}) = 1 + \Lambda_1 X_k^{-1} + \Lambda_2 X_k^{-2} + \cdots + \Lambda_\nu X_k^{-\nu} = 0$$

Multiplicando ambos os lados por $Y_k X_k^{j+\nu}$ (j é qualquer número tal que $1 \leq j \leq \nu$) obtemos

$$\begin{gathered} Y_k X_k^{j+\nu}(1 + \Lambda_1 X_k^{-1} + \Lambda_2 X_k^{-2} + \cdots + \Lambda_\nu X_k^{-\nu}) = 0 \\ \iff Y_k X_k^{j+\nu} + \Lambda_1 Y_k X_k^{j+\nu-1} + \Lambda_2 Y_k X_k^{j+\nu-2} + \cdots + \Lambda_\nu Y_k X_k^j = 0 \end{gathered}$$

Somando de $k = 1$ a ν, temos

$$\sum_{k=1}^{\nu}(Y_k X_k^{j+\nu} + \Lambda_1 Y_k X_k^{j+\nu-1} + \cdots + \Lambda_\nu Y_k X_k^j) = 0$$

$$\iff \left(\sum_{k=1}^{\nu} Y_k X_k^{j+\nu}\right) + \Lambda_1 \left(\sum_{k=1}^{\nu} Y_k X_k^{j+\nu-1}\right) + \cdots + \Lambda_\nu \left(\sum_{k=1}^{\nu} Y_k X_k^j\right) = 0$$

Note que estes somatórios são agora equivalentes aos valores da síndrome por (5.5), portanto

$$S_{j+\nu} + \Lambda_1 S_{j+\nu-1} + \cdots + \Lambda_{\nu-1} S_{j+1} + \Lambda_\nu S_j = 0$$

o que prova (i). □

Vamos acompanhar a prova acima no caso particular com $\delta = 7$ e $\nu = 3$. Note que $g(t) \mid p(t)$ e

$$g(t) = \prod_{j=1}^{6}(t - \alpha^j),$$

e neste caso teremos que

(i) Os coeficientes Λ_i do polinômio de localização do erro satisfazem o seguinte sistema linear

$$\begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} \begin{bmatrix} \Lambda_3 \\ \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_4 \\ -S_5 \\ -S_6 \end{bmatrix}$$

(ii) Os valores de erro Y_k satisfazem o seguinte sistema linear

$$\begin{bmatrix} X_1^1 & X_2^1 & X_3^1 \\ X_1^2 & X_2^2 & X_3^2 \\ X_1^3 & X_2^3 & X_3^3 \\ X_1^4 & X_2^4 & X_3^4 \\ X_1^5 & X_2^5 & X_3^5 \\ X_1^6 & X_2^6 & X_3^6 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{bmatrix}$$

Como $p(\alpha^j) = 0$ para $j = 1, 2, \ldots, 6$, temos

$$S_j = y(\alpha^j) = p(\alpha^j) + e(\alpha^j)$$

$$= e(\alpha^j) = \sum_{k=1}^{\nu} e_{i_k} \left(\alpha^j\right)^{i_k} = \sum_{k=1}^{6} Y_k X_k^j$$

Calculando as síndromes, temos

$$S_1 = Y_1X_1^1 + Y_2X_2^1 + Y_3X_3^1$$

$$S_2 = Y_1X_1^2 + Y_2X_2^2 + Y_3X_3^2$$

$$S_3 = Y_1X_1^3 + Y_2X_2^3 + Y_3X_3^3$$

$$S_4 = Y_1X_1^4 + Y_2X_2^4 + Y_3X_3^4$$

$$S_5 = Y_1X_1^5 + Y_2X_2^5 + Y_3X_3^5$$

$$S_6 = Y_1X_1^6 + Y_2X_2^6 + Y_3X_3^6$$

o que nos dá o item (ii). Mas,

$$\Lambda(x) = 1 + \Lambda_1 x + \Lambda_2 x^2 + \Lambda_3 x^3$$

onde $X_1^{-1}, X_2^{-1}, X_3^{-1}$ são as raízes de $\Lambda(x)$, assim, multiplicando ambos os lados das equações

$$\Lambda(X_1^{-1}) = 1 + \Lambda_1 X_1^{-1} + \Lambda_2 X_1^{-2} + \Lambda_3 X_1^{-3} = 0$$

$$\Lambda(X_2^{-1}) = 1 + \Lambda_1 X_2^{-1} + \Lambda_2 X_2^{-2} + \Lambda_3 X_2^{-3} = 0$$

$$\Lambda(X_3^{-1}) = 1 + \Lambda_1 X_3^{-1} + \Lambda_2 X_3^{-2} + \Lambda_3 X_3^{-3} = 0$$

respectivamente por $Y_1X_1^{1+3}$, $Y_2X_2^{1+3}$ e $Y_3X_3^{1+3}$, obtemos

$$Y_1X_1^4 + \Lambda_1 Y_1X_1^3 + \Lambda_2 Y_1X_1^2 + \Lambda_3 Y_1X_1 = 0$$

$$Y_2X_2^4 + \Lambda_1 Y_2X_2^3 + \Lambda_2 Y_2X_2^2 + \Lambda_3 Y_2X_2 = 0$$

$$Y_3X_3^4 + \Lambda_1 Y_3X_3^3 + \Lambda_2 Y_3X_3^2 + \Lambda_3 Y_3X_3 = 0$$

Somando as três equações, temos

$$(Y_1X_1^4 + Y_2X_2^4 + Y_3X_3^4) + \Lambda_1(Y_1X_1^3 + Y_2X_2^3 + Y_3X_3^3)$$
$$+ \Lambda_2(Y_1X_1^2 + Y_2X_2^2 + Y_3X_3^2) + \Lambda_3(Y_1X_1 + Y_2X_2 + Y_3X_3) = 0$$

Notemos que os coeficientes dos Λ_i são equivalentes aos valores das síndromes, assim

$$S_4 + \Lambda_1 S_3 + \Lambda_2 S_2 + \Lambda_3 S_1 = 0$$

e analogamente para as demais equações do item (i).

5.4.2 Exemplos

Exemplo 5.4.3. Como $\alpha = 3$ é um elemento primitivo de $\mathbb{F}_7$ podemos escolher o seguinte polinômio gerador:

$$g(t) = (t-3)(t-3^2)(t-3^3)(t-3^4) \in \mathbb{F}_7$$

$$g(t) = t^4 + 6t^3 + 3t^2 + 2t + 4$$

Como $\delta - 1 = 4$, logo $\delta = 5$ e o código corrige até 2 erros. Assim, temos a seguinte matriz geradora

$$G_{6\times 2} = \begin{bmatrix} 4 & 0 \\ 2 & 4 \\ 3 & 2 \\ 6 & 3 \\ 1 & 6 \\ 0 & 1 \end{bmatrix}$$

O polinômio e matriz paridade deste código são

$$h(t) = \frac{t^6 - 1}{t^4 + 6t^3 + 2t + 4} = t^2 + t + 5$$

$$H_{4\times 6} = \begin{bmatrix} 1 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 1 & 5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 5 & 0 \\ 0 & 0 & 0 & 1 & 1 & 5 \end{bmatrix}$$

Seja $y = c_1 + c_2$ uma palavra do código

$$y = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 6 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 4 \\ 2 \\ 3 \\ 6 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 5 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Temos

$$Hy = \begin{bmatrix} 1 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 1 & 5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 5 & 0 \\ 0 & 0 & 0 & 1 & 1 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 5 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$Hy = 0$, assim $y \in C$.

Vamos acrescentar um erro na palavra:

$$y = \begin{bmatrix} 4 \\ 6 \\ 5 \\ 2 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 6 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Como

$$Hy = \begin{bmatrix} 1 & 1 & 5 & 0 & 0 & 0 \\ 0 & 1 & 1 & 5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 5 & 0 \\ 0 & 0 & 0 & 1 & 1 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ 6 \\ 6 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

temos $Hy \neq 0$, então $y \notin C$.

Suponhamos que existem 2 erros.

Para $\nu = 2$. Vamos calcular as síndromes: como $y(t) = 4 + 6t + 6t^2 + 2t^3 + t^5$,

$$S_1 = y(\alpha^1) = 2 \qquad S_2 = y(\alpha^2) = 4$$
$$S_3 = y(\alpha^3) = 1 \qquad S_4 = y(\alpha^4) = 2$$

Para localizar os erros, resolvemos

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix} \tag{5.6}$$

Logo

$$\begin{bmatrix} 2 & 4 \\ 4 & 1 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \end{bmatrix} = \begin{bmatrix} 6 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 4 & 6 \\ 4 & 1 & 5 \end{bmatrix} \xrightarrow{L_2 \mapsto 5L_1 + L_2} \begin{bmatrix} 2 & 4 & 6 \\ 0 & 0 & 0 \end{bmatrix}$$

que é um sistema indeterminado, assim concluímos que $\nu < 2$.

Suponhamos então que existe um erro. Para $\nu = 1$:

$$S_1 \cdot \Lambda_1 = -S_2 \Longrightarrow 2 \cdot \Lambda_1 = -4 \Longrightarrow \Lambda_1 = -2 = 5$$

Substituindo no polinômio localizador de erro, temos $\Lambda(x) = 1 + x.\Lambda_1 = 1 + 5x$.

Temos candidatas a raízes $\alpha, \alpha^2, \alpha^3, \alpha^4$:

$$\Lambda(\alpha^1) = 2 \qquad \Lambda(\alpha^2) = 4$$
$$\Lambda(\alpha^3) = 3 \qquad \Lambda(\alpha^4) = 0$$

Desta forma, $\Lambda(\alpha^4) = 0$. Assim,

$$\frac{1}{X_1} = \alpha^4 \Longrightarrow X_1 = \alpha^{-4} = \alpha^2$$

Desta forma, encontramos que o erro está na posição 2.

$$e(t) = e_2 t^2$$
$$y(t) = p(t) + e_2 t^2$$

Temos: $y(\alpha) = e_2 \alpha^2$, como $\alpha^2 = 2$ e $y(\alpha) = 2$. De

$$2 = 2e_2 \Longrightarrow e_2 = 1,$$

encontramos que o valor do erro é 1.

Logo, $e(t) = 1 \cdot t^2$ e corrigimos: $p(t) = y(t) - e(t)$

$$\begin{bmatrix} 4 \\ 6 \\ 6 \\ 2 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 5 \\ 2 \\ 0 \\ 1 \end{bmatrix}$$

Exemplo 5.4.4. Seja $g(t) = 1 + 8t + 5t^2 + 3t^3 + t^4 \in \mathbb{F}_{11}$ do Exemplo 5.2.3, vamos encontrar uma palavra pertencente ao código fazendo

$$c_1 + 3c_3 = \begin{bmatrix} 1 \\ 8 \\ 5 \\ 3 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 3 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 8 \\ 5 \\ 3 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 8 \\ 5 \\ 5 \\ 9 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \in C.$$

Vamos colocar dois erros na mensagem de maneira que

$$y = \begin{bmatrix} 1 \\ 8 \\ 8 \\ 5 \\ 5 \\ 9 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 8 \\ 8 \\ 5 \\ 5 \\ 9 \\ 2 \\ 0 \\ 2 \\ 0 \end{bmatrix} \notin C.$$

Assim,

$$y(t) = 1 + 8t + 8t^2 + 5t^3 + 5t^4 + 9t^5 + 2t^6 + 2t^8 = g(t) + 3t^2 g(t) - t^6 + 2t^8$$

Temos $\alpha = 2$, então vamos calcular as síndromes e em seguida substituir na expressão do teorema 5.4.2(i):

$$S_1 = y(\alpha^1) = 8 \qquad S_2 = y(\alpha^2) = 3$$
$$S_3 = y(\alpha^3) = 7 \qquad S_4 = y(\alpha^4) = 3$$

Logo

$$\begin{bmatrix} 8 & 3 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -7 \\ -3 \end{bmatrix} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$$

Resolvendo o sistema encontramos $\Lambda_1 = -1$ e $\Lambda_2 = 5$, substituimos estes valores em

$$\Lambda(x) = (1 - xX_1)(1 - xX_2) = 1 + \Lambda_1 x^1 + \Lambda_2 x^2 = 1 + (-x) + 5(x^2)$$

cujas raízes são: $x = \dfrac{1}{X_1}$, $x = \dfrac{1}{X_2}$

Temos as seguintes candidatas a raízes de $\Lambda(x) = 5x^2 - x + 1$: $\alpha, \alpha^2, \ldots, \alpha^{10}$. Módulo 11, temos

$$\alpha = 2 \longrightarrow 5 \cdot 2^2 - 2 + 1 = 19 \neq 0$$
$$\alpha^2 = 4 \longrightarrow 5 \cdot 4^2 - 4 + 1 = 77 = 0$$
$$\alpha^4 = 5 \longrightarrow 5 \cdot 5^2 - 5 + 1 = 121 = 0$$
$$\alpha^6 = -2 \longrightarrow 5 \cdot (-2)^2 - (-2) + 1 = 1 \neq 0$$

Encontramos as raízes α^2 e α^4. Assim, $\Lambda(\alpha^2) = \Lambda(\alpha^4) = 0$. Portanto

$$\frac{1}{X_1} = \alpha^2 \Longrightarrow X_1 = \alpha^{-2} = \alpha^8 \qquad \text{e} \qquad \frac{1}{X_2} = \alpha^4 \Longrightarrow X_2 = \alpha^{-4} = \alpha^6$$

Desta forma, encontramos as posições dos erros, os quais encontram-se nas posições 6 e 8, assim, $e(t) = e_6 t^6 + e_8 t^8$. Precisamos encontrar o valor do erro, para isso, a partir de

$$y(t) = p(t) + e_6 t^6 + e_8 t^8$$

resolveremos o seguinte sistema:

$$\begin{cases} 8 = y(\alpha) = e_6 \alpha^6 + e_8 \alpha^8 \\ 3 = y(\alpha^2) = e_6 \alpha^{12} + e_8 \alpha^{16} \end{cases}$$

Como $\alpha^6 = -2 = 9$, $\alpha^8 = 3$, $\alpha^{12} = 4$ e $\alpha^{16} = 9$, temos:

$$\begin{cases} 8 = 9e_6 + 3e_8 \\ 3 = 4e_6 + 9e_8 \end{cases}$$

Resolvendo o sistema encontramos $e_6 = -1$ e $e_8 = 2$. Assim, encontramos o erro, $e(t) = -1t^6 + 2t^8$. E a mensagem original é

$$y(t) - e(t) = y(t) + t^6 - 2t^8$$

A seguir, veremos um exemplo com distância mínima $d(C) = 7$, podendo corrigir até três erros.

Observação 5.4.5. *Os cálculos foram feitas com auxílio do programa PARI/GP.*

Exemplo 5.4.6. Seja $\alpha = 6$ um elemento primitivo de $\mathbb{F}_{11}$, então o seguinte polinômio será o gerador do código de Reed-Solomon de parâmetros (10,4). A redundância do código cíclico será igual a 6, pois o grau$(g(t)) = 6$, enquanto a dim$(C) = n - \text{grau}(g(t)) = 10 - 6 = 4$.

$$\begin{aligned} g(t) &= (t-6)(t-6^2)(t-6^3)(t-6^4)(t-6^5)(t-6^6) \\ &= 6 + 3t + 8t^2 + 9t^3 + t^4 + 4t^5 + t^6 \end{aligned}$$

Assim, a matriz geradora é

$$G = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 3 & 6 & 0 & 0 \\ 8 & 3 & 6 & 0 \\ 9 & 8 & 3 & 6 \\ 1 & 9 & 8 & 3 \\ 4 & 1 & 9 & 8 \\ 1 & 4 & 1 & 9 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

E o polinômio e a matriz paridade é

$$h(t) = \frac{t^{10} - 1}{6 + 3t + 8t^2 + 9t^3 + t^4 + 4t^5 + t^6} = 9 + t + 4t^2 + 7t^3 + t^4$$

$$H = \begin{bmatrix} 1 & 7 & 4 & 1 & 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 4 & 1 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 7 & 4 & 1 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 7 & 4 & 1 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 7 & 4 & 1 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 7 & 4 & 1 & 9 \end{bmatrix}$$

Vamos encontrar uma palavra do código e adicionar dois erros:

$$2c_1 + c_3 = 2\begin{bmatrix}6\\3\\8\\9\\1\\4\\1\\0\\0\\0\end{bmatrix} + \begin{bmatrix}0\\0\\6\\3\\8\\9\\1\\4\\1\\0\end{bmatrix} = \begin{bmatrix}1\\6\\0\\10\\10\\6\\3\\4\\1\\0\end{bmatrix} \in C.$$

$$y = \begin{bmatrix}1\\6\\0\\10\\10\\6\\3\\4\\1\\0\end{bmatrix} + \begin{bmatrix}0\\0\\2\\0\\0\\0\\0\\0\\1\\0\end{bmatrix} = \begin{bmatrix}1\\6\\2\\10\\10\\6\\3\\4\\2\\0\end{bmatrix} \notin C.$$

$$\begin{aligned} y(t) &= 1 + 6t + 2t^2 + 10t^3 + 10t^4 + 6t^5 + 3t^6 + 4t^7 + 2t^8 \\ &= 2g(t) + t^2 g(t) + 2t^2 + t^8 \end{aligned}$$

Temos $\alpha = 6$, precisamos calcular as síndromes para descobrir quantos erros e a localização dos erros. Como

$$g(x) = \prod_{j=1}^{6}(x - \alpha^j),$$

calcularemos as seguintes síndromes $S(\alpha^i)$ para $i = 1, 2, \ldots, 6$:

$$S_1 = y(\alpha^1) = 10 \qquad S_2 = y(\alpha^2) = 1$$
$$S_3 = y(\alpha^3) = 8 \qquad S_4 = y(\alpha^4) = 0$$
$$S_5 = y(\alpha^5) = 3 \qquad S_6 = y(\alpha^6) = 10$$

Assume-se que o decodificador conhece o número de erros ν, e por tentativa, começando pelo maior valor de ν, iremos testar se o sistema possui solução. Neste caso, como $n-k \geq 2\nu$ e $n-k=6$, vamos resolver o seguinte sistema para $\nu=3$:

$$\begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} \begin{bmatrix} \Lambda_3 \\ \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_4 \\ -S_5 \\ -S_6 \end{bmatrix}$$

Substituindo os valores, temos

$$\begin{bmatrix} 10 & 1 & 8 \\ 1 & 8 & 0 \\ 8 & 0 & 3 \end{bmatrix} \begin{bmatrix} \Lambda_3 \\ \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \\ -10 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 1 \end{bmatrix}$$

Resolvendo o sistema

$$\begin{bmatrix} 10 & 1 & 8 & 0 \\ 1 & 8 & 0 & 8 \\ 8 & 0 & 3 & 1 \end{bmatrix} \xrightarrow{L_2 \mapsto L_1 + L_2} \begin{bmatrix} 10 & 1 & 8 & 0 \\ 0 & 9 & 8 & 8 \\ 8 & 0 & 3 & 1 \end{bmatrix}$$

$$\xrightarrow{L_3 \mapsto 8L_1 + L_3} \begin{bmatrix} 10 & 1 & 8 & 0 \\ 0 & 9 & 8 & 8 \\ 0 & 8 & 1 & 1 \end{bmatrix} \xrightarrow{L_3 \mapsto 4L_2 + L_3} \begin{bmatrix} 10 & 1 & 8 & 0 \\ 0 & 9 & 8 & 8 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Neste caso, uma linha da matriz é toda zero e o sistema é indeterminado, concluímos que o número de erros é menor e testaremos para $\nu = 2$ (como neste exemplo foram inseridos 2 erros já era sabido que $\nu = 2$), resolvendo o seguinte sistema:

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix}$$

Substituindo os valores, temos

$$\begin{bmatrix} 10 & 1 \\ 1 & 8 \end{bmatrix} \begin{bmatrix} \Lambda_2 \\ \Lambda_1 \end{bmatrix} = \begin{bmatrix} -8 \\ -0 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Resolvendo o sistema encontramos $\Lambda_1 = 4$ e $\Lambda_2 = 1$, substituindo estes valores no polinômio localizador de erro, temos

$$\Lambda(x) = 1 + \Lambda_1 x^1 + \Lambda_2 x^2 = 1 + 4x^1 + 1x^2$$

Precisamos encontrar as raízes do polinômio de localização de erro, para isso é necessário

fazer tentativa com as candidatas a raízes, as quais são: $\alpha, \alpha^2, \alpha^3, \cdots, \alpha^{10}$, sendo $\alpha = 6$.

$$\begin{aligned}
\alpha = 6 &\longrightarrow 1 + 4.6 + 6^2 = 61 = 6 \neq 0 \\
\alpha^2 = 3 &\longrightarrow 1 + 4.3 + 3^2 = 22 = 0 \\
\alpha^3 = 7 &\longrightarrow 1 + 4.7 + 7^2 = 78 = 1 \neq 0 \\
\alpha^4 = 9 &\longrightarrow 1 + 4.9 + 9^2 = 118 = 8 \neq 0 \\
\alpha^5 = 10 &\longrightarrow 1 + 4.10 + 10^2 = 141 = 9 \neq 0 \\
\alpha^6 = 5 &\longrightarrow 1 + 4.5 + 5^2 = 46 = 2 \neq 0 \\
\alpha^7 = 8 &\longrightarrow 1 + 4.8 + 8^2 = 97 = 2 \neq 0 \\
\alpha^8 = 4 &\longrightarrow 1 + 4.4 + 4^2 = 33 = 0
\end{aligned}$$

Desta forma, temos: $\Lambda(\alpha^2) = \Lambda(\alpha^8) = 0$. Assim,

$$\frac{1}{X_1} = \alpha^2 \Longrightarrow X_1 = \alpha^{-2} = \alpha^8 \qquad \text{e} \qquad \frac{1}{X_2} = \alpha^8 \Longrightarrow X_2 = \alpha^{-8} = \alpha^2$$

Desta forma, encontramos que os erros estão nas posições 2 e 8, assim, $e(t) = e_2 t^2 + e_8 t^8$.

Falta, então, encontrarmos o valor dos erros utilizando a seguinte equação:

$$y(t) = p(t) + e_2 t^2 + e_8 t^8$$

Temos

$$\begin{cases} 10 = y(\alpha) = e_2\alpha^2 + e_8\alpha^8 \\ 3 = y(\alpha^2) = e_2\alpha^4 + e_8\alpha^{16} \end{cases}$$

Como $\alpha^2 = 3$, $\alpha^4 = 9$, $\alpha^8 = 4$ e $\alpha^{16} = 5$, obtemos:

$$\begin{cases} 10 = 3e_2 + 4e_8 \\ 1 = 9e_2 + 5e_8 \end{cases}$$

Resolvendo o sistema temos: $e_2 = 2$ e $e_8 = 1$. Logo, $e(t) = 2t^2 + 1t^8$. Assim, a

mensagem original é

$$y(t) - e(t) = y(t) - 2t^2 - 1t^8 = \begin{bmatrix} 1 \\ 6 \\ 2 \\ 10 \\ 10 \\ 6 \\ 3 \\ 4 \\ 2 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ 0 \\ 10 \\ 10 \\ 6 \\ 3 \\ 4 \\ 1 \\ 0 \end{bmatrix}$$

6 CONCLUSÃO

Na perspectiva desta autora, a teoria dos códigos corretores de erros foi um campo de pesquisa desafiador, sobretudo sua aplicabilidade em trabalhos de alunos no Ensino Médio. Para tanto, foi necessário o aprofundamento e revisão dos diferentes conceitos de aritmética e álgebra linear. Outro desafio foi a necessidade de apropriação do conhecimento de softwares livres, tais como PARI/GP e Python.

Os algoritmos de codificação e decodificação de Códigos Lineares, principalmente os Códigos de Hamming, podem ser trabalhados no ensino médio como aplicação de operações com matrizes e resoluções de sistema lineares, enquanto os códigos cíclicos podem ser trabalhados no estudo de polinômios, sendo necessário o estudo de aritmética modular. Para os códigos cíclicos é indicado o uso de algum software, como por exemplo, o Python ou PARI/GP.

Seria interessante que na disciplina Recursos Computacionais no Ensino de Matemática do PROFMAT fossem trabalhados estes softwares.

REFERÊNCIAS

FERNANDEZ, C. d. S.; HEFEZ, A. **Introdução à Álgebra Linear**. Rio de Janeiro: SBM. (Coleção PROFMAT), 2016.

HAMMING, Richard. Disponível em: <http://sci-humor.blogspot.com/2014/01/you-and-your-research-lecture-by.html>. Acesso em: 11 julho. 2019.

HEFEZ, A.; VILLELA, M. L. T. **Códigos corretores de erros**. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 2008.

MARTINEZ, F. B. et al. **Teoria dos números: um passeio com primos e outros números familiares pelo mundo inteiro**. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 2010.

SHANNON, Claude. Disponível em: <https://pt.wikipedia.org/wiki/Claude_Shannon>. Acesso em: 12 julho. 2019.

TWUM, F. et al. Reed solomon encoding: Simplified explanation for programmers. **International Journal of Computer Science and Information Security**, LJS Publishing, v. 14, n. 12, p. 469, 2016.

VENTURA, J. **Notas de Combinatória e Teoria de Códigos**. 2011. Disponível em: <https://www.math.tecnico.ulisboa.pt/~jventura/CTC/ctc1415.html>. Acesso em: 28 nov. 2018.