UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUSTAVO MARCONDES DE LIMA

CÓDIGOS CORRETORES DE ERROS: FUNDAMENTOS TEÓRICOS E APLICAÇÕES

CURITIBA

2025

GUSTAVO MARCONDES DE LIMA

CÓDIGOS CORRETORES DE ERROS: FUNDAMENTOS TEÓRICOS E APLICAÇÕES

Error-correcting codes: theoretical foundations and applications

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Licenciado em Matemática do Curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná.
Orientadora: Prof^a. Dr^a. Patrícia Massae Kitani.

CURITIBA

2025

GUSTAVO MARCONDES DE LIMA

CÓDIGOS CORRETORES DE ERROS: FUNDAMENTOS TEÓRICOS E APLICAÇÕES

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Licenciado em Matemática do Curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 23 de junho de 2025

Patrícia Massae Kitani
Doutorado
Universidade Tecnológica Federal do Paraná

Ana Cristina Corrêa Munaretto
Doutorado
Universidade Tecnológica Federal do Paraná

Mari Sano
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA

2025

RESUMO

Este trabalho tem como objetivo dar fundamentação para o estudo dos códigos corretores de erros, fazendo uso de ferramentas de álgebra e de álgebra linear. Para isso, estudou-se alguns conceitos relacionados a corpos e algumas propriedades de espaços vetoriais. Assim, a primeira etapa deste trabalho consistiu na revisão dos conceitos básicos para avançar na teoria de códigos. Em seguida, conceitos novos, como distância mínima, métrica de Hamming e os parâmetros para definir códigos foram estudados. Essa preparação mostrou-se importante para uma análise mais aprofundada de códigos lineares, os quais desempenham um papel fundamental em uma variedade de aplicações em eletrônica, comunicações e armazenamento de dados, destacando-se pela sua notável habilidade em identificar e corrigir erros.

Palavras-chave: códigos corretores de erros; códigos lineares; corpos finitos; espaço vetorial.

ABSTRACT

This work aims to provide a basis for the study of error-correcting codes, making use of tools of algebra and linear algebra. For this, some concepts related to fields and some properties of vector spaces were studied. Thus, the first stage of this work consisted of the revision of the basic concepts to advance in the theory of codes. In sequence, new concepts, like minimum distance, Hamming metric, and the parameters to define codes, were studied. This preparation proved to be important for a more in-depth analysis of linear codes, which play a fundamental role in a variety of applications in electronics, communications, and data storage, highlighting their notable ability in identifying and correcting errors.

Keywords: error-correcting codes; linear codes; finite fields; vector space.

SUMÁRIO

1 INTRODUÇÃO

Se pararmos para refletir por um momento sobre o cotidiano, podemos concluir que com os adventos da modernidade, os sistemas de comunicação se tornaram cada vez mais sofisticados, aumentando de forma significativa a praticidade do envio de informações. Tais adventos foram possíveis graças ao desenvolvimento de teorias matemáticas que serviram de suporte aos sistemas de comunicação.

Vale ressaltar que quando falamos em envio de informações nessa área de estudo, devemos considerar qualquer meio de transmissão que recorra à recursos computacionais, como por exemplo: transmissão de dados por satélite, transmissão por Wi-fi, telecomunicações e até mesmo nossas simples mensagens de texto.

O que não está explícito é que por trás de toda essa troca de informações, existe um sistema complexo. Sempre que uma mensagem é enviada através de um canal, nesse canal existem interferências e perturbações, chamadas de ruídos. Esses ruídos causam modificações nas mensagens, possibilitando que a mesma seja interpretada erroneamente pelo receptor. Com o uso da álgebra linear para construção dos Códigos Corretores de Erros, conseguimos fazer que a mensagem transmitida pelo canal chegue em seu destino da forma mais confiável possível.

Segundo Sabadin (2019) descoberta dessa área de estudo se deu por volta dos anos 1950, com as publicações dos trabalhos de Shannon e Hamming, ambos matemáticos e ambos funcionários da empresa Bell Labs (empresa de pesquisa industrial e científica, de propriedade da empresa Nokia). Claude Elwood Shannon (1916-2001) publicou em 1948 o artigo "*A mathematical theory of communication*", dando marco à area denominada Teoria de Informações, onde a teoria de códigos tem grande presença. Logo depois, Richard Wesley Hamming (1915-1998) publica o "*Código de Hamming*", criando um algoritmo que estrutura um código corretor de erro. O trabalho foi finalizado poucos anos antes, mas por questões burocráticas relacionadas a patentes, Hamming só conseguiu realizar a publicação do mesmo em 1950.

Nessa época era alto o investimento para comprar um computador, sendo limitado principalmente à universidades e centros de pesquisas. Era na Bell Labs que Hamming tinha acesso à essas máquinas. Restritamente nos fins de semana, que era quando ele levava seus dados para serem lidos. Os computadores tinham um sistema de detecção de erros desses dados, mas não de correção. Quando o sistema encontrava um erro, encerrava a leitura dos dados e passava a ler os dados do próximo usuário. Dessa forma Hamming perdeu duas semanas de serviço, tendo em vista que ele só poderia verificar se o computador fez a leitura completa do seu material uma semana depois de ter dado inicio à leitura. A necessidade de criar um instrumento que fosse além, um instrumento que fosse capaz não só de detectar um erro mas também corrigí-lo, foi o que deu motivação ao matemático.

Também com grande relevância, Marcel Jules Edouard Golay (1902-1989), que tinha acesso a computadores na empresa de tecnologia, onde trabalhava, estendeu os resultados das publicações de Shannon e Hamming em seu artigo "*Notes on Digital Coding*". Nesse artigo

ele desenvolve o Código de Golay, um código corretor de erros que também é utilizado nos dias de hoje.

Esses três matemáticos foram os que deram início ao desenvolvimento da área de Teoria de Códigos, por volta dos anos 50. No entanto, com a popularização dos computadores e com a corrida espacial desencadeada pela Guerra Fria (1947-1991), após a década de 60 a Teoria de Códigos passou a ser estudada e desenvolvida também por engenheiros.

Por exemplo, em 1965, a NASA utilizou a nave espacial *Mariner 4* para enviar 22 imagens, cada uma com uma resolução de 200×200 *pixels*, onde seriam atribuídos uma das 64 tonalidades pré-estabelecidas para cada *pixel*. No entanto, como a capacidade de processamento dos computadores eram menores, as imagens eram apenas codificadas, sem passarem por algum código corretor de erro. Dessa forma ao serem transmitidas pelo canal, os ruídos possibilitavam alterações na tonalidade de cinza dos *pixels*, causando distorções na imagem original introduzida no canal. Em resumo, as imagens recebidas nem sempre refletiam fielmente a coloração exata que havia sido enviada.

Sete anos depois, a *Mariner 9* transmitiu novas imagens de Marte, com uma resolução bem melhor de 700×832 *pixels*. O desenvolvimento dos computadores tornou o processamento mais rápido e também tornou viável a codificação e a passagem por um código corretor de erros, corrigindo informações da imagem que foram modificadas pelos ruídos. Neste caso, o código usado foi o de Reed-Muller, um caso particular dos códigos lineares.

Já em 1979 com um pouco mais de avanço computacional, a nave *Voyager* fez transmissões de imagens coloridas do Planeta Júpiter, na qual além da codificação, foi utilizado o Código da Golay para uma codificação corretora de erro. A passagem pelo Código de Golay corrigiu os ruídos causados pelo canal de transmissão, fazendo também com que a imagem chegasse ao destino da forma mais nítida.

A cada ano que passa, com os progressos da tecnologia, as imagens têm sido cada vez mais nítidas.

O principal objetivo deste projeto consiste na investigação dos códigos corretores de erros, com um foco especial nos códigos lineares. Os códigos lineares são amplamente empregados em sistemas de comunicação e na preservação de dados. São conhecidos pela sua habilidade em identificar e corrigir erros, tornando-os fundamentais em diversas aplicações. Os exemplos desses códigos são: códidos de Hamming; BCH; Reed-Muller e Reed-Solomon. A escolha do código vai depender dos objetivos e das necessidades de sua aplicação. Estudaremos a estrutura básica desse código sem adentrar no estudo de cada código linear.

Além dos códigos lineares, existem várias outras classes de códigos que são usados nos dias de hoje. Não será o objeto deste trabalho, mas citamos alguns dos códigos mais comuns: os códigos de Barra; códigos QR; de Acesso e Autenticação; de Correção de Erros Não Lineares; de Correção de Erros para Armazenamento; de Criptografia e Códigos de Compressão de Dados.

No Capítulo 1 damos a fundamentação teórica de alguns assuntos tratados durante as disciplinas de álgebra e álgebra linear. Somente os tópicos necessários como corpos finitos,

espaços e subespaços vetoriais, base, dimensão e produto interno serão abordados. As referências bibliográficas para esse capítulo foram o livro de *Um Curso de Álgebra Linear* (Coelho e Lourenço, 2013) e as notas de aula de Ventura (2013). No Capítulo 2 damos uma breve introdução aos códigos corretores de erros, apresentando os elementos que definem códigos. As referências utilizadas foram: *Códigos Corretores de Erros* (Hefez e Villela, 2008), *Breve introdução à teoria dos códigos corretores de erros* (Millies, 2009) e, *Notas de Combinatória e Teoria de Códigos* (Ventura, 2013). No Capítulo 3 abordamos códigos lineares, resultados e definições que os estruturam e como se dá o processo de detecção e correção de erros através dos mesmos. A principal referência para esse capítulo foi *Códigos Corretores de Erros* (Hefez e Villela, 2008). Para o desenvolvimento de certos resultados, utilizou-se como referência o artigo *Um primeiro curso sobre códigos corretores de erros* (Bahia, 2010).

2 CONCEITOS PRÉVIOS

Antes de adentrarmos no estudo da Teoria de Códigos, é fundamental compreender alguns conceitos preliminares. Como essa teoria se desenvolve sobre estruturas algébricas, especialmente corpos finitos, é necessário um entendimento prévio dessas estruturas. Além disso, noções de Álgebra Linear — como espaços vetoriais e transformações lineares, principalmente — também desempenham um papel essencial na construção e na análise dos códigos. Os conceitos apresentados neste capítulo servirão, portanto, como base para o estudo formal da Teoria de Códigos.

Neste trabalho os conjuntos $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}, \mathbb{R}$ e $\mathbb{C}$ denotarão o conjunto dos números naturais, inteiros, racionais, reais e complexos, respectivamente.

2.1 Anéis, anéis de integridade e corpos.

Definição 2.1. *Seja $\mathbb{A}$ um conjunto não vazio munido de duas operações $+$ e $\cdot$, que chamaremos de adição e multiplicação:*

$$\begin{array}{rcccl} + & : & \mathbb{A}\times\mathbb{A} & \to & \mathbb{A} \\ & & (x,y) & \mapsto & x+y \end{array} \qquad \begin{array}{rcccl} \cdot & : & \mathbb{A}\times\mathbb{A} & \to & \mathbb{A} \\ & & (x,y) & \mapsto & x\cdot y \end{array}$$

*$(\mathbb{A}, +, \cdot)$ é dito **anel** se para quaisquer $a, b, c \in \mathbb{A}$, verifica-se as seis seguintes condições.*

1. $(a+b)+c = a+(b+c)$ *(Associatividade da adição).*
2. *Existe um elemento $0_{\mathbb{A}} \in \mathbb{A}$ tal que $a + 0_{\mathbb{A}} = 0_{\mathbb{A}} + a = a$ (Existência do elemento neutro para a adição).*
3. *Existe $x \in \mathbb{A}$ $(x = -a)$, tal que $a + x = x + a = 0_{\mathbb{A}}$ (Existência do inverso aditivo).*
4. $a + b = b + a$ *(Comutatividade da adição).*
5. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ *(Associatividade da multiplicação).*
6. $a \cdot (b + c) = a \cdot b + a \cdot c$ *(Distributividade à esquerda da multiplicação em relação à adição).*
   $(a+b) \cdot c = a \cdot c + b \cdot c$ *(Distributividade à direita da multiplicação em relação à adição).*

Vejamos alguns exemplos de conjuntos com estrutura de anel.

Exemplo 2.2. $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ *e* $\mathbb{C}$ *são exemplos clássicos de conjuntos, com as operações $+$ e $\cdot$ usuais, que têm estrutura de anel. Nesses casos, o número zero é o elemento neutro para a adição.*

Note que o conjunto $\mathbb{N}$ não tem estrutura de anel já que para qualquer $a \in \mathbb{N}^*$, tem-se que não existe $x \in \mathbb{N}$ tal que $a + x = x + a = 0_{\mathbb{N}}$. As outras condições são verificadas.

Exemplo 2.3. *O conjunto* $2\mathbb{Z} = \{2k \mid k \in \mathbb{Z}\}$ *dos inteiros pares, munido das operações* $+$ *e* $\cdot$ *usuais tem estrutura de anel.*

Nesse exemplo, primeiro verificaremos que o conjunto é fechado para adição e para a multiplicação, isto é, dados x,$y \in 2\mathbb{Z}$ teremos $x + y \in 2\mathbb{Z}$ e $x \cdot y \in 2\mathbb{Z}$.

Tomemos $x = 2a, y = 2b \in 2\mathbb{Z}$, a,$b \in \mathbb{Z}$, então $x + y = 2a + 2b = 2(a + b)$. Isso implica que $x + y \in 2\mathbb{Z}$.

Da mesma forma com o produto, $x \cdot y = 2a \cdot 2b = 2(a2b)$. Assim, a operação $\cdot$ também é fechada em $2\mathbb{Z}$.

Portanto, as operações de soma e produto por escalar são fechadas em $2\mathbb{Z}$.

Agora vamos verificar que $(2\mathbb{Z}, +, \cdot)$ tem estrutura de anel.

1. Associatividade da adição:

   Sejam $x = 2a, y = 2b,\ z = 2c \in 2\mathbb{Z}$, a,b,$c \in \mathbb{Z}$.

   Como x,y,$z \in \mathbb{Z}$ e a operação de adição é a mesma em $2\mathbb{Z}$ e $\mathbb{Z}$, então vale a associatividade para x,y,$z \in 2\mathbb{Z} \subseteq \mathbb{Z}$. Logo, $(x + y) + z = x + (y + z)$.

2. Existência do elemento neutro para a adição:

   Tome $0 = 2 \cdot 0 \in 2\mathbb{Z}$. Assim, para todo $x \in 2\mathbb{Z}$ tem-se $x + 0 = 0 + x = x$, uma vez que 0 é o elemento neutro da adição em $\mathbb{Z}$ e $x \in 2\mathbb{Z} \subset \mathbb{Z}$.

   Logo, existe o neutro para adição em $2\mathbb{Z}$.

3. Existência do inverso aditivo:

   Sejam $x = 2a, y = 2b \in 2\mathbb{Z}$, com $a, b \in \mathbb{Z}$, e suponhamos que $x + y = 0$. Logo $2a + 2b = 0$ e então, $2b = -(2a)$. Assim $y = -x \in 2\mathbb{Z}$.

   Logo, para todo $x \in 2\mathbb{Z}$, existe $y = -x \in 2\mathbb{Z}$ de forma que $x + y = 0$.

   As propriedades comutativa da adição, associativa da multiplicação e a distributiva da multiplicação em relação à adição são verificadas de imediato, já que $2\mathbb{Z} \subseteq \mathbb{Z}$.

   Logo, $x \cdot (y + z) = x \cdot y + x \cdot z$, para todo x,y,$z \in 2\mathbb{Z}$.

   E, portando, $(2\mathbb{Z}, +, \cdot)$ tem estrutura de anel.

Antes de citar os próximos exemplos de anéis, que serão fundamentais para nosso estudo, faz-se imprescindível a apresentação sobre um conceito aritmético. Brevemente trataremos sobre congruência.

Definição 2.4. *Sejam* a,$b, m \in \mathbb{Z}, m > 0$. *Dizemos que* a ***é congruente a*** b ***módulo*** m*, se* m *divide* $(a - b)$.

Denotaremos por $m|(a - b)$ *para sinalizar que* m *divide* $(a - b)$*;*

Usaremos a notação $a \equiv b\, mod\, m$ *para sinalizar que* a *é congruente a* b *módulo* m*. Se* a *e* b *não forem congruentes módulo m, dizemos que* a *e* b *são incongruentes módulo m.*

Exemplo 2.5. *Alguns casos de congruência:*

a) $5 \equiv 17\, mod\, 3,$ *pois* $3|(5-17)$*;*

b) $25 \equiv 41\, mod\, 4,$ *pois* $4|(25-41)$*;*

c) $33 \equiv 89\, mod\, 7,$ *pois* $7|(33-89)$.

A Definição 2.4 estabelece uma relação entre dois elementos de $\mathbb{Z}$, denominada *congruência módulo* m.

Para essa relação, valem as seguintes propriedades:

Propriedades 2.6. *Sejam* a,b, *e* c *números inteiros. Temos:*

1. $a \equiv a\, mod\, m$ *(Reflexiva);*
2. $a \equiv b\, mod\, m \Rightarrow b \equiv a\, mod\, m$ *(Simétrica);*
3. $a \equiv b\, mod\, m$ *e* $b \equiv c\, mod\, m \Rightarrow a \equiv c\, mod\, m$ *(Transitiva).*

Demonstração.

1. Se $a \equiv a\, mod\, m$, então da definição de congruência, $m|(a-a)$. De fato, pois $m|0$. Logo, é válida a propriedade reflexiva.

2. Seja $a \equiv b\, mod\, m$. Da definição de congruência, $m|(a-b)$. Com isso e lembrando que a e b são números inteiros, temos então que $m|(b-a)$, portanto, $b \equiv a\, mod\, m$. Logo, vale a simetria.

3. Sejam $a \equiv b\, mod\, m$ e $b \equiv c\, mod\, m$. Da definição, $m|(a-b)$ e $m|(b-c)$. Isso significa que $(a-b) = mq_1$ e $(b-c) = mq_2$, para certos q_1,q_2 inteiros. Reescrevendo a segunda equação, obtemos $b = mq_2 + c$. Agora basta substituir $b = mq_2 + c$ na primeira igualde. Assim, temos $(a - mq_2 - c) = mq_1$ e então, $(a-c) = mq_1 + mq_2$. Por fim, $(a-c) = m(q_1+q_2)$. Com isso, $m|(a-c)$ e portanto, $a \equiv c\, mod\, m$. Logo, é válida a propriedade transitiva.

□

Desta forma, a relação de congruência módulo m sobre $\mathbb{Z}$ é uma relação de equivalência.

Definição 2.7. *A* ***classe de equivalência de um número inteiro*** a *é o conjunto* $\{x \in \mathbb{Z} | x \equiv a\, mod\, m\}$, *e será denotada por* $\overline{a}$*. O conjunto de todas as classes de equivalência de uma congruencia módulo m é denotada por* $\mathbb{Z}/m\mathbb{Z}$ *ou* $\mathbb{Z}_m$*.*

Assim, dois números inteiros são congruentes módulo m se, e somente se, pertencem à mesma classe de equivalência.

A próxima proposição apresenta uma importante perspectiva no estudo de congruência.

Proposição 2.8. *Para quaisquer a e b números inteiros, $a \equiv b \, mod \, m$ se, e somente se, a e b tem o mesmo resto na divisão euclidiana por m.*

Demonstração.

($\Rightarrow$) Levemos em consideração que a divisão euclidiana de b por m seja escrita por $b = mk + r$ para algum k inteiro, onde r é o resto dessa divisão e $0 \leqslant r < m$. Por hipótese, $m|(a - b)$ e, portanto, $a = qm + b$, para algum q inteiro. Das duas igualdades, obtemos então que $a = qm + (mk + r)$ e consequentemente, $a = m(q + k) + r$. Mas isso significa que r é o resto da divisão euclidiana de a por m.

Logo, a e b tem o mesmo resto na divisão euclidiana por m.

($\Leftarrow$) Da hipótese, $a = mq_1 + r$ e $b = mq_2 + r$, com $0 \leqslant r < m$. Sendo assim, $(a - b) = mq_1 + r - mq_1 - r$ e consequentemente, $(a - b) = m(q_1 - q_2)$. Isso implica que $m|(a - b)$. Logo, $a \equiv b \, mod \, m$.

□

Definição 2.9. *Um conjunto de m inteiros, $m > 0$, forma um **sistema completo de restos módulo** m se, dados quaisquer dois elementos distintos desse conjunto, eles forem incongruentes módulo m.*

Exemplo 2.10. *O conjunto $\{6, 7, 8, 9, 10, 11\}$ é um sistema completo de restos módulo 6.*

Exemplo 2.11. *O conjunto $\{0, 1, 2, 3, ..., m - 1\}$ é um sistema completo de restos módulo m. Note que esse conjunto tem m elementos.*

Proposição 2.12. *Se $\{r_1, r_2, r_3, ..., r_m\}$ é um sistema completo de restos módulo m, então qualquer inteiro a é congruente a um e somente um dos $r_i, 0 < i \leq m$.*

Demonstração.

Realizando a divisão euclidiana de a por m, obtemos $a = mq + r$, $0 \leq r < m$, para algum q inteiro. Com isso, temos que $a \equiv r \, mod \, m$. Como consequência da Proposição 2.8, ao dividirmos $r_i, 0 < i \leq m$, obtemos restos diferentes dois a dois. Portanto, existe um r_i tal que $r_i = mq_i + r$, ou seja, $r_i \equiv r \, mod \, m$ e por transitividade, $r_i \equiv a \, mod \, m$. Se $a \equiv r_k \, mod \, m$, então $r_i \equiv r_k \, mod \, m$ e da definição de sistema completo de restos, concluímos que $r_i = r_k$.

□

Como consequência desta proposição, temos que $\mathbb{Z}_m = \left\{\overline{0}, \overline{1}, \overline{2}, ..., \overline{m-1}\right\}$. Vamos definir as operações $+$ e $\cdot$ para $\mathbb{Z}_m$ da seguinte forma:

$$\begin{array}{rccc} +: & \mathbb{Z}_m \times \mathbb{Z}_m & \longrightarrow & \mathbb{Z}_m \\ & (\overline{a}, \overline{b}) & \longmapsto & \overline{a} + \overline{b} = \overline{a+b} \end{array} \qquad \begin{array}{rccc} \cdot: & \mathbb{Z}_m \times \mathbb{Z}_m & \longrightarrow & \mathbb{Z}_m \\ & (\overline{a}, \overline{b}) & \longmapsto & \overline{a} \cdot \overline{b} = \overline{ab} \end{array}$$

Mostraremos que estas operações estão bem definidas, isto é, que independe do representante da classe de equivalência.

Sejam $(\overline{a}, \overline{b}) = (\overline{c}, \overline{d}) \in \mathbb{Z}_m \times \mathbb{Z}_m$.

$(\overline{a}, \overline{b}) = (\overline{c}, \overline{d})$ se, e somente se, $\overline{a} = \overline{c}$ e $\overline{b} = \overline{d}$. Mas então isso significa que $a \equiv c \, mod \, m$ e $b \equiv d \, mod \, m$.

Assim $a = mq_1 + c$ e $b = mq_2 + d, q_1, q_2 \in \mathbb{Z}$. Então:

1. $a+b = mq_1+c + mq_2+d = (c+d)+m(q_1+q_2)$. Assim $a+b \equiv c+d \, mod \, m$, ou seja, $\overline{a+b} = \overline{c+d}$, concluindo que a adição está bem definida.

2. $ab = (mq_1+c)(mq_2+d) = m(q_1mq_2+q_1d+cq_2)+cd,$ portanto, $ab \equiv cd \, mod \, m$. Logo $\overline{ab} = \overline{cd}$, o que mostra que a operação de multiplicação também está bem definida.

Exemplo 2.13. *Sejam* $\overline{2}, \overline{3} \in \mathbb{Z}_6$. *Aqui* $\overline{2} + \overline{3} = \overline{2+3} = \overline{5}$ *e* $\overline{2} \cdot \overline{3} = \overline{6} = \overline{0}$.

Exemplo 2.14. *: O conjunto* $\mathbb{Z}_6 = \{\overline{0},\overline{1},\overline{2},\overline{3},\overline{4},\overline{5}\}$ *com as operações de adição e multiplicação definidas anteriormente tem estrutura de anel.*

As condições 1,4,5 e 6 da definição de anel são fáceis de serem verificadas diretamente. Sendo assim, vamos verificar as condições 2 e 3.

Seja $\overline{0} \in \mathbb{Z}_6$. Note que $\overline{x} + \overline{0} = \overline{x}$, para todo $\overline{x} \in \mathbb{Z}_6$. Além disso, $\overline{0} + \overline{0} = \overline{0}$, portanto, $\overline{0} = -(\overline{0})$.

Ou seja, além da classe $\overline{0}$ ser o elemento neutro da soma, é também seu próprio inverso aditivo. Vamos definir quem são os inversos aditivos dos outros elementos de $\mathbb{Z}_6$.

$\overline{1} + \overline{x} = \overline{0}$ se, e somente se, $\overline{1+x} = \overline{0}$. Isso implica que $\overline{x} = \overline{5}$, logo, $\overline{5}$ é o inverso aditivo de $\overline{1}$. Da mesma forma que $\overline{1}$ é o inverso aditivo de $\overline{5}$.

$\overline{2} + \overline{x} = \overline{0}$ se, e somente se, $\overline{2+x} = \overline{0}$. Assim $\overline{x} = \overline{4}$, logo, $\overline{4}$ é o inverso aditivo de $\overline{2}$. Do mesmo modo que $\overline{2}$ é o inverso aditivo de $\overline{4}$.

$\overline{3} + \overline{x} = \overline{0}$ se, e somente se, $\overline{3+x} = \overline{0}$. Isso mostra que $\overline{x} = \overline{3}$, portanto, $\overline{3}$ é o inverso aditivo de $\overline{3}$.

Note que para todo $\overline{x} \in \mathbb{Z}_6$, existe um único $-(\overline{x}) = \overline{6-x} \in \mathbb{Z}_6$ tal que $\overline{x}+(-\overline{(x)}) = \overline{0}$.

De modo geral, temos que o conjunto $\mathbb{Z}_m = \{\overline{0}, \ \overline{1}, \ \overline{2}, \ \overline{3}, \cdots, \overline{m-2}, \ \overline{m-1}\}$, com $m \in \mathbb{N}$, e $-\overline{x} = \overline{m-x}$, para todo $x \in \mathbb{Z}_m$, tem estrutura de anel.

Definição 2.15. *Seja* $(\mathbb{A}, +, \cdot)$ *um anel que satisfaz a condição:*

$$\exists 1_{\mathbb{A}} \in \mathbb{A}, \ 1_{\mathbb{A}} \neq 0_{\mathbb{A}} : \ 1_{\mathbb{A}} \cdot x = x \cdot 1_{\mathbb{A}} = x, \ \forall x \in \mathbb{A}.$$

Neste caso dizemos que $(\mathbb{A}, +, \cdot)$ *é* ***um anel com unidade.***

Exemplo 2.16. $(\mathbb{Z}, +, \cdot)$ *com as operações usuais é anel com unidade.*

Buscando facilitar a leitura, utilizaremos a notação 0 para nos referirmos ao elemento neutro e 1 quando se tratar da unidade de um anel, se não houver confusão.

Definição 2.17. *Se* $(\mathbb{A}, +, \cdot)$ *é um anel que satisfaz a seguinte condição:*

$$x \cdot y = y \cdot x, \ \forall x,y \in \mathbb{A},$$

então $(\mathbb{A}, +, \cdot)$ *é* ***um anel comutativo.***

Exemplo 2.18. $\mathbb{Z}$ *e* $\mathbb{Z}_m$ *são anéis comutativo.*

Definição 2.19. *Se* $(\mathbb{A}, +, \cdot)$ *é um anel que satisfaz a condição:*

$$x \cdot y = 0 \Leftrightarrow x = 0 \text{ ou } y = 0, x,y \in \mathbb{A},$$

então dizemos que $(\mathbb{A}, +, \cdot)$ *é* ***um anel sem divisores de zero.*** *Caso contrário, dizemos que é um anel com divisores de zero.*

Se $x \cdot y = 0$*, com* $x \neq 0$ *e* $y \neq 0$*, dizemos que* x *e* y *são divisores de zero.*

O conjunto $\mathbb{Z}_6 = \{\overline{0},\overline{1},\overline{2},\overline{3},\overline{4},\overline{5}\}$ *do Exemplo 2.14 citado anteriormente, é um anel que contém divisores de zero. Basta notar que* $\overline{2} \cdot \overline{3} = \overline{6} = \overline{0}$*. Ou seja, existem* $\overline{x} \neq \overline{0}$ *e* $\overline{y} \neq \overline{0}$ *em* $\mathbb{Z}_6$ *tais que* $\overline{x} \cdot \overline{y} = \overline{0}$.

Finalmente, podemos apresentar a definição de anel de integridade.

Definição 2.20. *Se* $(\mathbb{A}, +, \cdot)$ *é um anel com unidade e sem divisores de zero, dizemos que* $(\mathbb{A}, +, \cdot)$ *é um* ***anel de integridade*** *ou* ***domínio de integridade****.*

No caso do Exemplo 2.2, os conjuntos $(\mathbb{Z}, +, \cdot)$, $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$ e $(\mathbb{C}, +, \cdot)$ também são exemplos de conjuntos, munidos pelas operações $+$ e $\cdot$ conhecidas usualmente, que têm estrutura de anel de integridade.

No Exemplo 2.3, o conjunto dos inteiros pares $2\mathbb{Z} = \{2k \mid k \in \mathbb{Z}\}$, munido das operações $+$ e $\cdot$ usuais tem estrutura de anel, é anel comutativo e anel sem divisores de zero, mas não é um domínio de integridade, já que não existe $1_{2\mathbb{Z}} \in 2\mathbb{Z}$ que satisfaz $x \cdot 1_{2\mathbb{Z}} = x$, para todo $x \in 2\mathbb{Z}$.

Vamos supor por absurdo que $2\mathbb{Z}$ é anel com unidade, ou seja, existe $1_{2\mathbb{Z}} \in 2\mathbb{Z}$ que satisfaça $x \cdot 1_{2\mathbb{Z}} = x$, para todo $x \in 2\mathbb{Z}$, $x \neq 0$.

Como $x, 1_{2\mathbb{Z}} \in 2\mathbb{Z}$ podemos escrever $1_{2\mathbb{Z}} = 2a$ e $x = 2b$ com $a,b \in \mathbb{Z}$. Então:

$x \cdot 1_{2\mathbb{Z}} = x \Rightarrow 2b \cdot 2a = 2b \Rightarrow (2b \cdot 2a) - (2b) = 0 \Rightarrow 2b(2a - 1) = 0.$

Sendo $2\mathbb{Z}$ um anel comutativo sem divisores de zero, segue que $2b(2a - 1) = 0$, se, e somente se, $2b = 0$ ou $(2a - 1) = 0$.

Da hipótese temos $2b \neq 0$, então resta que $2a - 1 = 0$, e consequentemente $2a = 1$. Uma contradição, pois $1 \notin 2\mathbb{Z}$ já que não pode ser escrito como múltiplo de 2.

Logo, $2\mathbb{Z}$ não é anel com unidade. Consequentemente, não é anel de integridade.

No Exemplo 2.14, o conjunto $\mathbb{Z}_6 = \{\overline{0},\overline{1},\overline{2},\overline{3},\overline{4},\overline{5}\}$ é anel comutativo com unidade mas não é domínio de integridade, pois como citado anteriormente, os elementos $\overline{2}$ e $\overline{3}$ são divisores de zero.

De modo geral não podemos dizer de modo geral que o conjunto $\mathbb{Z}_m = \{\overline{0},\overline{1},\overline{2},...,\overline{m-2},\overline{m-1}\}$, é ou não domínio de integridade para qualquer que seja o valor m.

Ao definirmos estrutura de corpos posteriormente, também iremos definir quais são os valores de m que satisfarão $\mathbb{Z}_m$ ser domínio de integridade.

Definição 2.21. *Seja* $(\mathbb{K}, +, \cdot)$ *um anel de integridade ou domínio de integridade, que satisfaz a seguinte condição:*

$$\forall\, x \in \mathbb{K}\backslash\{0\},\, \exists\, y \in \mathbb{K} \,|\, x \cdot y = 1.$$

Ou seja, todo elemento de $\mathbb{K}\setminus\{0\}$ *têm inverso multiplicativo. Dessa forma dizemos que* $(\mathbb{K},+,\cdot)$ *tem estrutura de* ***Corpo algébrico****.*

Voltando mais uma vez ao Exemplo 2.2, os conjuntos $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}, +, \cdot)$ e $(\mathbb{C}, +, \cdot)$, com as operações $+$ e $\cdot$ usuais, continuam sendo exemplos de conjuntos, que além de serem domínios de integridade, também têm estrutura de corpo. Nesse caso $(\mathbb{Z}, +, \cdot)$ não se enquadra. Basta verificar que, para qualquer $x \in \mathbb{Z}\backslash\{0,1,-1\}$, seu inverso multiplicativo dado por $y = x^{-1} = \dfrac{1}{x} \notin \mathbb{Z}$.

Já no Exemplo 2.3, o conjunto $2\mathbb{Z} = \{2k \,|\, k \in \mathbb{Z}\}$ dos inteiros pares, munido das operações $+$ e $\cdot$ usuais não é domínio de integridade, consequentemente não tem estrutura de corpo.

No Exemplo 2.14 o conjunto $\mathbb{Z}_6 = \{\overline{0},\overline{1},\overline{2},\overline{3},\overline{4},\overline{5}\}$ tem divisores de zero, consequentemente não tem estrutura de corpo.

Exemplo 2.22. $\mathbb{Z}_3 = \{\overline{0}, \overline{1}, \overline{2}\}$ *é corpo pois* $\overline{1} \cdot \overline{1} = \overline{1}$ *e* $\overline{2} \cdot \overline{2} = \overline{1}$.

Vimos acima que o conjunto $\mathbb{Z}_6$ não é corpo algébrico e $\mathbb{Z}_3$ é. Dessa forma, fica muito amplo dizer que o conjunto $\mathbb{Z}_m = \{\overline{0},\overline{1},\overline{2},...,\overline{m-2},\overline{m-1}\}, m \in \mathbb{N}$, tem estrutura de corpo para qualquer que seja o valor m. Para construirmos esse raciocínio, precisamos de alguns conceitos de aritmética modular.

Buscando verificar quais os valores de m que satisfazem $\mathbb{Z}_m$ ser corpo, vamos analisar m subdividindo em dois casos: m não é primo e m é primo.

No caso de m não ser primo, podemos escrever m como produto de dois naturais r e s, da seguinte forma: $m = r \cdot s,\ r < m,\ s < m$. Sendo assim:

$$\overline{m} = \overline{0} \Rightarrow \overline{r \cdot s} = \overline{0} \Rightarrow \overline{r} \cdot \overline{s} = \overline{0}.$$

Isso quer dizer que $\overline{r}$ e $\overline{s}$ são divisores próprios de zero. Com isso verificamos que $\mathbb{Z}_m = \mathbb{Z}_{r\cdot s}$ não pode ser anel de integridade e consequentemente não é corpo.

Já quando m é primo, a única forma de escrevê-lo como produto de dois naturais é $1 \cdot m$. Assim temos que $\overline{0} = \overline{m} = \overline{1 \cdot m}$ e, portanto, $\overline{m \cdot q} = \overline{m} \cdot \overline{q} = \overline{0}$ para todo $q \in \mathbb{N}$.

Concluímos que x pertence à classe do zero em $\mathbb{Z}_m$ se, e somente se x é múltiplo de m, ou seja, se existe $q \in \mathbb{N}$ tal que $x = m \cdot q$.

Essa verificação servirá de base para enunciar a proposição seguinte.

Proposição 2.23. *O conjunto $\mathbb{Z}_m$ é corpo, se, e somente se, m é primo.*

Demonstração.

Vamos supor que $\mathbb{Z}_m$ é um corpo e que m não é primo. Então podemos escrever m como sendo produto de dois números, r e s naturais: $m = rs$ com $r,s \in \mathbb{N}$. Isso implica que $\overline{m} = \overline{rs} = \overline{r}\ \overline{s} = \overline{0}$, ou seja, $\overline{r}$ e $\overline{s}$ são divisores próprios de $\overline{0}$. Uma contradição. Logo, m é primo.

Agora, suponha que m seja primo e sejam $a,b \in \mathbb{Z}_m$ tais que $\overline{a}\overline{b} = \overline{0}$. Assim $ab \in \overline{0}$, que equivale a dizer que $ab = mq$, para algum $q \in \mathbb{Z}$. Como m é primo, então m divide a ou m divide b, ou seja, $a = mx$ ou $b = my$, com $x,y \in \mathbb{Z}$. Portanto, $\overline{a} = \overline{0}$ ou $\overline{b} = \overline{0}$.

Mostraremos agora que todo elemento não nulo possui inverso multiplicativo, ou seja, para todo $\overline{a} \neq 0$, existe $\overline{b}$ tal que $\overline{a} \cdot \overline{b} = \overline{1}$. Como m é primo, para todo número a, $1 \leq a \leq m$, temos que a e m são primos entre si, ou seja, $mdc(a,m) = 1$. Pelo Teorema de Bézout, existem inteiros x e y tais que $ax + my = 1$. Aplicando a congruência módulo m, temos que $\overline{1} = \overline{ax} + \overline{my} = \overline{ax} + \overline{0} = \overline{ax}$. Ou seja, a classe $\overline{x}$ é o inverso multiplicativo de $\overline{a}$ em $\mathbb{Z}_m$, logo, todo elemento não nulo tem inverso. Além disso, temos que a adição e a multiplicação são comutativas e associativas em $\mathbb{Z}_m$. Logo, $\mathbb{Z}_m$ é corpo.

□

Esse resultado é muito importante para nossos estudos de Códigos, pois quando operamos nessa área de conhecimento, trabalhamos justamente com espaços vetoriais sobre o corpo dos $\mathbb{Z}_p$, p primo. A seção seguinte irá esclarecer a relação desses dois elementos.

2.2 Espaços vetoriais e subespaços vetoriais

Um espaço vetorial também é uma estrutura algébrica, constituída por dois elementos: um corpo $\mathbb{K}$, denominado corpo de escalares, e um conjunto não vazio de vetores que usualmente denotaremos por V.

Definição 2.24. *Seja V um conjunto, $\mathbb{K}$ um corpo e duas operações, $+$ sobre dois elementos de V e $\cdot$ de um escalar de $\mathbb{K}$ sobre um elemento de V, isto é*

$$\begin{array}{rcccl} + & : & V \times V & \to & V \\ & & (v,w) & \mapsto & v+w \end{array} \qquad \begin{array}{rcccl} \cdot & : & \mathbb{K} \times V & \to & V \\ & & (\lambda,v) & \mapsto & \lambda \cdot v \end{array}$$

*Diremos que $V \neq \emptyset$ é um **Espaço vetorial** sobre o corpo $\mathbb{K}$ se para quaisquer $u,v,w \in V$ e quaisquer $\lambda, \mu \in \mathbb{K}$, verifica-se as seguintes condições:*

1. $(u+v)+w = u+(v+w)$.
2. *Existe um elemento $0 \in V$ tal que $u+0 = 0+u = u$, para todo $u \in V$.*

3. *Para cada* $v \in V$*, existe um elemento em* V*, denotado por* $-v$*, tal que* $v + (-v) = -v + v = 0$.

4. $u + v = v + u$.

5. $\lambda \cdot (u + v) = \lambda \cdot u + \lambda \cdot v$.

6. $(\lambda + \mu) \cdot u = \lambda \cdot u + \mu \cdot u$.

7. $(\lambda\mu) \cdot u = \lambda \cdot (\mu \cdot u)$.

8. $1 \cdot u = u$.

Os elementos de um espaço vetorial são chamados vetores. O vetor do Item 2 é dito vetor nulo e $-v$ é dito oposto de v.

O espaço vetorial de nosso interesse é o seguinte: sendo p um número primo, o conjunto $\mathbb{Z}_p^n\,(n \in \mathbb{N})$ é um espaço vetorial sobre o corpo $\mathbb{Z}_p$, onde dados $\vec{u} = (u_1,...,u_n)$ e $\vec{v} = (v_1,...,v_n) \in \mathbb{Z}_p^n$ e $\lambda \in \mathbb{Z}_p$, as operações são definidas da forma usual:

$$\begin{array}{rcccl} + & : & \mathbb{Z}_p^n \times \mathbb{Z}_p^n & \to & \mathbb{Z}_p^n \\ & & (\vec{v},\vec{w}) & \to & \vec{v} + \vec{w} \end{array} \qquad \begin{array}{rcccl} \cdot & : & \mathbb{Z}_p \times \mathbb{Z}_p^n & \to & \mathbb{Z}_p^n \\ & & (\lambda,\vec{v}) & \to & \lambda \cdot \vec{v} \end{array}$$

$\vec{v} + \vec{w} = (v_1,...,v_n) + (w_1,...,w_n) = (v_1 + w_1,...,v_n + w_n)$ e
$\lambda \cdot \vec{v} = \lambda \cdot (v_1,...,v_n) = (\lambda \cdot v_1,...,\lambda \cdot v_n)$;

Mostraremos que a Condição 1 vale para o espaço definido, ou seja, vamos verificar que dados $\vec{u} = (u_1,...,u_n)$, $\vec{v} = (v_1,...,v_n)$ e $\vec{w} = (w_1,...,w_n)$ pertencentes a $\mathbb{Z}_p^n$, tem-se $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$.

Para isso, basta notar que

$$\begin{aligned} (\vec{u} + \vec{v}) + \vec{w} &= (u_1 + v_1, \ldots, u_n + v_n) + (w_1, \ldots, w_n) \\ &= ((u_1 + v_1) + w_1, \ldots, (u_n + v_n) + w_n) \\ &= (u_1 + (v_1 + w_1), \ldots, u_n + (v_n + w_n)) \\ &= (u_1, \ldots, u_n) + (v_1 + w_1, \ldots, v_n + w_n) = \vec{u} + (\vec{v} + \vec{w}). \end{aligned}$$

Logo, $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$.

Facilmente podemos verficar as outras condições, concluindo assim que o conjunto $\mathbb{Z}_p^n$ munido das operações que foram definidas, é um espaço vetorial.

Definição 2.25. *Seja* V *um espaço vetorial sobre um corpo de escalares* $\mathbb{K}$*. Um subconjunto não vazio* $W \subset V$ *é dito* ***subespaço vetorial*** *de* V*, se* W *também é um espaço vetorial sobre o mesmo corpo de escalares* $\mathbb{K}$*, munido das mesmas operações definidas em* V*.*

Dado um espaço vetorial V sobre um corpo $\mathbb{K}$ e um subconjunto W contido nesse espaço, para verificarmos se W é subespaço vetorial de V precisamos verificar se todas as condições que definem um espaço vetorial são satisfeitas em W. No entanto, como $W \subset V$ temos de imediato que algumas condições são verificadas. A condição 1 (associatividade) e a condição 4 (comutatividade), por exemplo, não precisamos demonstrá-las, pois como são válidas em V, automaticamente são válidas para todos $u,v,w \in W$. O que faltaria mostrar é que W é fechado para adição e para multiplicação por um escalar.

Em resumo, se $W \neq \emptyset$ é fechado para a operação de adição e para a multiplicação pelo escalar de $\mathbb{K}$, que estão definidas sobre V, podemos dizer com certeza que W é um subespaço vetorial de V. Isso será melhor esclarecido na proposição seguinte.

Proposição 2.26. *Seja V um espaço vetorial sobre um corpo $\mathbb{K}$ e seja $\emptyset \neq W \subset V$. W é subespaço vetorial de V se, e somente se, as seguintes condições são verificadas:*

i) $u + v \in W$, *para todo* $u,v \in W$*;*

ii) $\alpha \cdot u \in W$, *para todo* $\alpha \in \mathbb{K}$ *e todo* $u \in W$*.*

Demonstração.

Por definição W é espaço vetorial. Logo são verificadas as condições *i)* e *ii)*.

Agora vamos provar que W satisfaz todas as condições de espaço vetorial tomando as condições *i)* e *ii)* como hipóteses.

Como W é subconjunto de V e além disso *i)* e *ii)* nos dizem que as operações são fechadas para adição de vetores e para a multiplicação pelo escalar do corpo $\mathbb{K}$ por um vetor de W, então as condições 1, 4, 5, 6, 7 e 8 da definição de espaço vetorial são verificadas de imediato. Sendo assim, para provar que W é espaço vetorial basta apenas provar as condições 2 e 3.

Já que $W \neq \emptyset$, existe $u \in W$. Além disso, da condição *ii)* nós temos que $\alpha \cdot u \in W,$ para todo $\alpha \in \mathbb{K}$. Em particular, para $\alpha = 0$, teremos que $\alpha \cdot u = 0 \cdot u = \vec{0} \in W$. Assim $\vec{0} + u = u + \vec{0} = u$, para todo $u \in W$.

Da mesma forma, para $\alpha = (-1) \in \mathbb{K}$, $\alpha \cdot u = (-1) \cdot u = -(u) \in W$.

Assim, W é espaço vetorial. □

Um ponto que vale ser citado, é a diferença entre o escalar 0 que pertence à $\mathbb{K}$ e o vetor nulo, que pertence ao subespaço W. Notemos que, ao multiplicar um vetor por um escalar, o resultado é um vetor, sendo assim, na demonstração acima fez-se necessário sinalizar que $\alpha \cdot u = \vec{0}$, para que fique claro ao leitor essa diferença. Mas não havendo confusão, denotaremos o vetor nulo somente por 0.

Exemplo 2.27. *Seja V um espaço vetorial. Note que o conjunto $\{0\}$ satisfaz as condições de subespaço de V. Da mesma forma, V também é um subespaço vetorial de V. Esses dois subespaços são denominados os subespaços triviais do espaço V.*

Definição 2.28. *Seja V um espaço vetorial sobre um corpo de escalares $\mathbb{K}$ e seja $v \in V$ um vetor qualquer. Dizemos que v é* ***combinação linear*** *dos vetores $v_1,v_2,...,v_n \in V$ se existem escalares $\alpha_1,\alpha_2,...,\alpha_n \in \mathbb{K}$ de modo que*

$$v = \alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + ... + \alpha_n \cdot v_n.$$

Em outros termos, dizemos que o vetor $v \in V$ é escrito em função dos vetores $v_1,v_2,...,v_n$ de V multiplicados pelos escalares de $\mathbb{K}$.

Exemplo 2.29. *O elemento $v = (3,2) \in \mathbb{R}^2$ é combinação linear dos vetores $\beta_1 = (1,0)$ e $\beta_2 = (0,1)$.*

De fato, perceba que $3 \cdot \beta_1 + 2 \cdot \beta_2 = 3 \cdot (1,0) + 2 \cdot (0,1) = (3,2)$.

Exemplo 2.30. *O elemento $v = (-5,4,0) \in \mathbb{R}^3$ é combinação linear dos vetores $\beta_1 = (1,0,0), \beta_2 = (0,1,0), \beta_3 = (0,0,1)$.*

De fato, $-5 \cdot \beta_1 + 4 \cdot \beta_2 + 0 \cdot \beta_3 = -5 \cdot (1,0,0) + 4 \cdot (0,1,0) + 0 \cdot (0,0,1) = (-5,4,0)$.

Exemplo 2.31. *Considere β_i um vetor tal que a i-ésima entrada do vetor é 1 e as demais são 0. De modo geral, todo vetor $v = (v_1,v_2,...,v_n) \in \mathbb{R}^n$ é combinação linear dos n vetores $\beta_1 = (1,0,...,0), \beta_2 = (0,1,...,0), ..., \beta_n = (0,0,...,1)$.*

Definição 2.32. *Seja V um espaço vetorial e seja $S = \{s_1, s_2,..., s_n\} \subset V$. Dizemos que o conjunto S é um* ***conjunto gerador*** *de V se todo elemento de V pode ser escrito como combinação linear dos elementos de S.*

Definição 2.33. *Dado um conjunto de vetores $\{v_1,v_2,...,v_n\} \subset V$ e $\alpha_1,\alpha_2,...,\alpha_n \in \mathbb{K}$, dizemos que esse conjunto de vetores é* ***Linearmente Independente (LI)*** *se a combinação linear $\alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + ... + \alpha_n \cdot v_n = 0$ implicar em $\alpha_1 = \alpha_2 = ... = \alpha_n = 0$.*

Agora, se a equação possui uma solução não trivial, ou seja, existe $\alpha_i \neq 0$ para algum $\alpha_i \in \mathbb{K}, 1 \leq i \leq n$, significa que podemos escrever ao menos um dos vetores como combinação linear dos demais vetores.

De fato, suponha que o escalar $\alpha_m \neq 0, m \in \{1, 2, \ldots, n\}$. Então

$$\alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + ... + \alpha_{m-1} \cdot v_{m-1} + \alpha_m \cdot v_m + \alpha_{m+1} \cdot v_{m+1} + ... + \alpha_n \cdot v_n = 0$$

$$\Leftrightarrow \alpha_1 \cdot v_1 + \alpha_2 \cdot v_2 + ... + \alpha_{m-1} \cdot v_{m-1} + \alpha_{m+1} \cdot v_{m+1} + ... + \alpha_n \cdot v_n = -(\alpha_m \cdot v_m)$$

$$\Leftrightarrow (\frac{-\alpha_1}{\alpha_m}) \cdot v_1 + (\frac{-\alpha_2}{\alpha_m}) \cdot v_2 + ... + (\frac{-\alpha_{m-1}}{\alpha_m}) \cdot v_{m-1} + ... + (\frac{-\alpha_n}{\alpha_m}) \cdot v_n = v_m.$$

Neste caso, como $\{v_1,v_2,...,v_n\} \subset V$ não satisfazem a definição de serem linearmente independentes, dizemos que eles são **Linearmente Dependentes (LD)**.

Exemplo 2.34. *Sejam os vetores $u = (3,5)$ e $v = (6,10)$ do espaço vetorial $\mathbb{R}^2$ sobre o corpo $\mathbb{R}$. Podemos escrever $v = 2u$, em outras palavras, um pode ser escrito em função do outro. Dizemos então que v é combinação linear de u.*

Outra forma de interpretar é verificar que $\alpha_1 u + \alpha_2 v = (0,0)$ *tem solução com* $\alpha_1, \alpha_2 \neq 0$. *Basta assumir* $\alpha_1 = -2$ *e* $\alpha_2 = 1$ *que assim temos* $-2 \cdot (3,5) + 1 \cdot (6,10) = (0,0)$.

Exemplo 2.35. *O conjunto de vetores* $\{(2,6,4), (-4,-4,2), (-6,-2,8)\}$ *do espaço vetorial* $\mathbb{R}^3$ *sobre* $\mathbb{R}$ *é L.D. pois* $1 \cdot (2,6,4) + 2 \cdot (-4,-4,2) - 1 \cdot (-6,-2,8) = (0,0,0)$.

Definição 2.36. *Seja* V *um espaço vetorial sobre um corpo* $\mathbb{K}$. *Dizemos que um subconjunto* $\beta \subset V$ *é uma* ***base*** *de* V *se:*

1. *Os vetores de* β *forem linearmente independentes.*
2. *Os vetores de* β *geram todo o espaço* V. *Em outras palavras, todo elemento de* V *pode ser escrito como combinação linear dos vetores que compõe o subconjunto* β.

Para um subconjunto β ser uma base de um espaço V, não basta que esse subconjunto apenas gere todo o espaço V, mas ainda é preciso que os vetores que compõe esse subconjunto sejam linearmente independentes.

No Exemplo 2.29, o conjunto $\beta = \{\beta_1, \beta_2\} = \{(1,0), (0,1)\}$ é linearmente independente, uma vez que $\alpha_1 \cdot (1,0) + \alpha_2 \cdot (0,1) = (0,0)$ se, e somente se, $\alpha_1 = \alpha_2 = 0$. Note também que todo vetor de $\mathbb{R}^2$ pode ser escrito como combinação linear do conjunto β. Assim, β é uma base de $\mathbb{R}^2$.

Da mesma forma no Exemplo 2.30. Todo vetor de $\mathbb{R}^3$ pode ser escrito como combinação linear do conjunto $\beta = \{\beta_1, \beta_2, \beta_3\} = \{(1,0,0), (0,1,0), (0,0,1)\}$, linearmente independente. Assim, β é uma base de $\mathbb{R}^3$.

Nestes dois exemplos, as bases são as denominadas bases canônicas de $\mathbb{R}^2$ e $\mathbb{R}^3$.

Definição 2.37. *Dizemos que um espaço vetorial* V *sobre um corpo* $\mathbb{K}$ *é* ***finitamente gerado*** *se o conjunto de vetores que geram* V *é finito.*

Teorema 2.38. *Seja* V *um espaço vetorial e* $S = \{v_1, v_2, \ldots, v_n\}$ *um conjunto finito de vetores que geram* V. *Então, é possível extrair um subconjunto de* S *que é uma base para* V.

Demonstração.

Sendo S um conjunto finito de vetores que gera V, qualquer elemento de V pode ser escrito como uma combinação linear de vetores de S.

Se os vetores de S forem linearmente independentes, então o subconjunto é o próprio S, ou seja, nada precisa ser feito.

Agora, vamos considerar que os vetores de S são linearmente dependentes. Isso significa que existe algum elemento x de S que pode ser escrito como combinação linear dos demais. Sendo assim, definimos $S_1 = S \setminus \{x\} \subseteq S$. Note que S_1 continua gerando V, pois como x é combinação linear dos demais vetores de S, então qualquer combinação linear que envolva x, pode ser reescrita como combinação linear apenas dos elementos de S_1. Se S_1 for linearmente independente, então S_1 seria uma base de V. Caso contrário, repetimos o processo

até eliminarmos todos os elementos que são escritos como combinação linear dos demais elementos de S_i. Isso é possivel, pois estamos assumindo que V é finitamente gerado. Assim, determinamos $B \subseteq S$ tal que B ainda gera V e B é linearmente independente.

□

Teorema 2.39. *Seja V um espaço vetorial finitamente gerado por um conjunto com n elementos. Então, qualquer conjunto linearmente independente em V possui no máximo n elementos.*

Demonstração.

Suponha que exista um conjunto $W = \{w_1, w_2, \ldots, w_m\} \subseteq V$, com $m > n$. Vamos mostrar que esse conjunto é linearmente dependente.

Sabemos que V é gerado por n vetores. Pelo Teorema 2.39, é possível extrair desses vetores uma base de V. Seja então $\{v_1, v_2, \ldots, v_r\}$ uma base de V, com $r \leq n$.

Como $\{v_1, \ldots, v_r\}$ é uma base, cada elemento w_j de W, $1 \leq j \leq m$, pode ser escrito como combinação linear dos vetores da base. Ou seja, existem $\alpha_{1j}, \ldots, \alpha_{rj}$ tais que

$$w_j = \alpha_{1j}v_1 + \alpha_{2j}v_2 + \cdots + \alpha_{rj}v_r. \tag{1}$$

Agora, consideramos uma combinação linear dos vetores de W igualando ao vetor nulo:

$$\beta_1 w_1 + \beta_2 w_2 + \cdots + \beta_m w_m = 0. \tag{2}$$

Substituindo a igualdade (1) na igualdade (2), temos a equação

$$\beta_1(\alpha_{11}v_1 + \cdots + \alpha_{r1}v_r) + \cdots + \beta_m(\alpha_{1m}v_1 + \cdots + \alpha_{rm}v_r) = 0.$$

Agrupando os v_j e os colocando em evidência, temos que

$$(\beta_1\alpha_{11} + \cdots + \beta_m\alpha_{1m})v_1 + \cdots + (\beta_1\alpha_{r1} + \cdots + \beta_m\alpha_{rm})v_r = 0.$$

Como $\{v_1, \ldots, v_r\}$ é uma base, ele é linearmente independente. Ou seja, a igualdade acima será igual a zero se, e somente se, os coeficientes forem iguais a zero, como mostra no seguinte sistema de equação:

$$\begin{cases} \beta_1\alpha_{11} + \cdots + \beta_m\alpha_{1m} = 0 \\ \vdots \\ \beta_1\alpha_{r1} + \cdots + \beta_m\alpha_{rm} = 0 \end{cases}$$

Temos um sistema com mais variáveis do que equações, ou seja, o sistema admite solução não trivial. Portanto, existe uma combinação linear não trivial de $w_1, \ldots, w_m$ que resulta no vetor nulo. Logo, esses vetores são linearmente dependentes.

□

Teorema 2.40. *Seja V um espaço vetorial finitamente gerado sobre o corpo $\mathbb{K}$ e seja β uma base qualquer de V. Temos que:*

1. *Cada elemento de V é escrito de maneira única como combinação linear da base β.*

2. *Qualquer base de V é formada pela mesma quantidade de vetores.*

Demonstração.

1. Da hipótese, β é uma base de V, vamos considerar $\beta = \{v_1,...,v_n\}$. Da Definição 2.36, temos que todo elemento de V pode ser escrito como combinação linear dos elementos de β.

Agora provaremos a unicidade. Suponha que podemos escrever o vetor de dois modos, digamos $u = x_1 \cdot v_1 + ... + x_n \cdot v_n = y_1 \cdot v_1 + ... + y_n \cdot v_n$. Isso vai implicar que $x_1 \cdot v_1 + ... + x_n \cdot v_n - y_1 \cdot v_1 - ... - y_n \cdot v_n = 0$ e, portanto, $(x_1 - y_1)v_1 + ... + (x_n - y_n)v_n = 0$. Sendo os elementos de β linearmente independentes, $(x_1-y_1)v_1+...+(x_n-y_n)v_n = 0$ ocorre somente se $x_1 = y_1, ...,x_n = y_n$, logo, todo elemento de V se escreve de forma única como combinação linear dos elementos de β.

2. Sejam β e β' duas bases de V. Como V é finitamente gerado, da Definição 2.37, temos que ambas as bases são formadas por uma quantidade finita de vetores, digamos m e m'. Sendo β um gerador de V, linearmente independente, temos do Teorema 2.39 que $m' \leq m$. Da mesma forma, considerando β' um gerador, linearmente independente, temos que $m \leq m'$. Portanto, $m = m'$.

□

Agora, traremos um corolário que decorre imediatamente do Teorema anterior. Mas antes, tomando como partida o item 2, vamos definir a dimensão de um espaço vetorial.

Definição 2.41. *Seja β uma base de um espaço vetorial V. A* ***dimensão*** *de V, denotada por $dim\, V$, é definida como a quantidade de vetores de β, ou seja, a quantidade de elementos que compõe qualquer base do espaço vetorial. Se o espaço vetorial for o trivial $V = \{0\}$, então $dim\, V = 0$.*

Exemplo 2.42. *Seja o espaço $\mathbb{R}^3$ sobre o corpo $\mathbb{R}$. Todo vetor desse espaço pode ser escrito como combinação linear dos vetores $\{(1,0,0); (0,1,0); (0,0,1)\}$ que são linearmente independentes. O conjunto $\beta = \{(1,0,0); (0,1,0); (0,0,1)\}$ com três elementos, gera todo o espaço $\mathbb{R}^3$. Assim $dim\, \mathbb{R}^3 = 3$.*

Corolário 2.43. *Seja V um espaço de dimensão $n \geq 1$ e seja β um subconjunto de V com n elementos. As seguintes afirmações são equivalentes:*

1. *β é uma base.*

2. *β é linearmente independente.*

3. *β é um conjunto gerador de V.*

Proposição 2.44. *Seja W um subespaço vetorial de um espaço vetorial finitamente gerado V sobre um corpo $\mathbb{K}$. Se $dim\, W = dim\, V$ então $W = V$.*

Demonstração.

Vamos supor que $dim\ W = dim\ V = n$. Seja $\beta = \{w_1,w_2,...,w_n\}$ uma base de W. Vamos supor por absurdo que $W \neq V$. Como $W \subsetneq V$, então existe $v \in V$ tal que $v \notin W$. Desse modo, o conjunto $\{w_1,w_2,...,w_n,v\} \subset V$ é linearmente independente. Como este conjunto tem $n+1$ vetores e $dim\, V = n$, obtemos uma contradição. Logo, $W = V$.

□

A próxima definição é a de forma bilinear simétrica, que é a mesma do produto interno usual quando o espaço vetorial é sobre o corpo dos reais ou complexo. Geralmente, na literatura matemática, o produto interno é definido para espaços vetoriais sobre o corpo dos números reais ou o corpo dos números complexos, que não é o nosso caso. Para o caso de um corpo qualquer, é necessário um prévio estudo de módulos. Para o nosso estudo, que envolve corpo finito, empregaremos unicamente a forma bilinear simétrica, a qual é convencionalmente designada como produto interno na maioria das referências bibliográficas sobre códigos. Além disso, no caso dos códigos trataremos de espaços vetoriais do tipo $\mathbb{K}^n$ sobre $\mathbb{K}$, então existirá uma base canônica (análogo à base cânônica do $\mathbb{R}^n$) e assim cada elemento deste espaço vetorial pode ser escrito como $u = (u_1, u_2, \cdots, u_n)$ $= u_1e_1 + u_2e_2... + u_ne_n$, onde e_i é o vetor com todas as entradas nulas e a i-ésima entrada é a unidade de $\mathbb{K}$. Os próximos resultados envolvendo produto interno serão tratados para este tipo de espaço vetorial.

Definição 2.45. *Seja V um espaço vetorial sobre um corpo $\mathbb{K}$, finitamente gerado. Uma **forma bilinear simétrica em** V (ou produto interno), é uma aplicação que relaciona dois vetores $u = (u_1,\ u_2,...,u_n)$ e $v = (v_1,\ v_2,...,v_n)$ de V a um elemento de $\mathbb{K}$, $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{K}$, operados da seguinte maneira:*

$$\langle u,v\rangle = u_1v_1 + u_2v_2 + ... + u_nv_n.$$

Uma conta simples mostra que:

1. $\langle u,v\rangle = \langle v,u\rangle$.
2. $\langle u + \alpha w, v\rangle = \langle u,v\rangle + \alpha\langle w,v\rangle$.

Definição 2.46. *Dois vetores $u,v \in V$ são ditos **ortogonais**, denotado por $u \perp v$, quando o produto interno entre eles resulta no escalar $0 \in \mathbb{K}$, ou seja, $\langle u, v\rangle = 0$.*

Definição 2.47. *Dado um subespaço vetorial $W \subset V$ munido de produto interno, o complemento ortogonal de W é o conjunto $W^{\perp} = \{v \in V \mid \langle v,w\rangle = 0,\ \forall\, w \in W\}$.*

Teorema 2.48. *Seja W um subespaço de V. Então o complemento ortogonal $W^{\perp}$ também será um subespaço vetorial de V.*

Demonstração.

Sejam w e 0 pertencentes a W. $\langle w,0\rangle = 0$ para todo w. Portanto, 0 pertence a $W^{\perp}$.

Tomemos u e v pertencentes a $W^{\perp}$. Então $\langle u+v,w\rangle = \langle u,w\rangle + \langle v,w\rangle = 0+0=0$. Portanto, a soma é fechada em $W^{\perp}$.

Seja $u \in W^{\perp}$ e seja $\alpha \in \mathbb{K}$. Assim $\langle \alpha u,w\rangle = \alpha\langle u,w\rangle = \alpha 0 = 0$. Portanto, a multiplicação por escalar é fechada em $W^{\perp}$.

Logo, $W^{\perp}$ é subespaço vetorial de V.

□

Os resultados apresentados a seguir são imprescindíveis para o estudo de uma categoria específica de Códigos Corretores de Erros, os Códigos Lineares. Em resumo, essa categoria de código é bem estruturada usando como suporte os resultados que se seguem. Antes de tudo, começaremos definindo o que é uma transformação linear.

Definição 2.49. *Sejam U e V espaços vetoriais sobre um mesmo corpo $\mathbb{K}$ e seja $T : U \to V$ uma função. T será dita uma **Transformação Linear** se*

$T(u_1+u_2) = T(u_1)+T(u_2)$ *para todos* u_1,u_2 *pertencentes a* U*;*
$T(\alpha u) = \alpha T(u)$*, para todo* α *pertencente a* $\mathbb{K}$ *e* u *pertencente a* U*.*

Exemplo 2.50. *A função identidade $T : U \to U$ é uma transformação linear.*

Exemplo 2.51. *Seja $\alpha \in \mathbb{R}$. A função $T_\alpha : \mathbb{R} \to \mathbb{R}$ dada por $T_\alpha(x) = \alpha x$, $x \in \mathbb{R}$, é uma transformação linear.*

Tomemos dois elementos x_1,x_2 pertencentes a $\mathbb{R}$. Assim $T_\alpha(x_1+x_2) = \alpha(x_1+x_2)$. Como estamos operando números reais, temos que $\alpha(x_1+x_2) = \alpha x_1+\alpha x_2 = T_\alpha(x_1)+T_\alpha(x_2)$. Logo, $T_\alpha(x_1+x_2) = T_\alpha(x_1)+T_\alpha(x_2)$. Seja agora $\beta \in \mathbb{R}$, então $T_\alpha(\beta x) = \alpha(\beta x)$. Se tratando de números reais, temos então que $\alpha(\beta x) = \beta(\alpha x) = \beta T_\alpha(x)$.

Exemplo 2.52. *A função $T : \mathbb{R}^2 \to \mathbb{R}^2$ dada por $T(x,y) = (y,x)$ é uma transformação linear. Sejam $u = (x_1,y_1), v = (x_2,y_2)$ pertencentes a $\mathbb{R}^2$.*

$T(u+v) = T(x_1+x_2,y_1+y_2) = (y_1+y_2,x_1+x_2) = (y_1,x_1)+(y_2,x_2) = T(u)+T(v)$.
Seja agora $\alpha \in \mathbb{R}$. Então $T(\alpha u) = T(\alpha(x,y)) = T(\alpha x,\alpha y) = (\alpha y,\alpha x) = \alpha(y,x) = \alpha T(u)$.

Exemplo 2.53. *A função $T : \mathbb{R}^2 \to \mathbb{R}^2$ dada por $T(x,y) = (x,0)$ é uma transformação linear. Sejam $u = (x_1,y_1), v = (x_2,y_2)$ pertencentes a $\mathbb{R}^2$.*

$T(u+v) = T(x_1+x_2,y_1+y_2) = (x_1+x_2,0) = (x_1,0)+(x_2,0) = T(u)+T(v)$.
Se $\alpha \in \mathbb{R}$, então $T(\alpha u) = T(\alpha x_1,\alpha y_1) = (\alpha x_1,0) = \alpha(x_1,0) = \alpha T(u)$.

Exemplo 2.54. *A função $T : \mathbb{Z}_2^2 \to \mathbb{Z}_2^3$ dada por $T(x,y) = (x+y,x,y)$ é uma transformação linear. Sejam $u = (x_1,y_1), v = (x_2,y_2)$ pertencentes a $\mathbb{Z}_2^2$.*

$$T(u+v) = T(x_1+x_2,y_1+y_2) = (x_1+x_2+y_1+y_2, x_1+x_2,y_1+y_2)$$
$$= (x_1+y_1, x_1,y_1)+(x_2+y_2, x_2,y_2) = T(u)+T(v).$$

Seja agora α pertencente a $\mathbb{Z}_2$. Então

$$T(\alpha u) = T(\alpha(x,y)) = T(\alpha x,\alpha y) = (\alpha x + \alpha y, \alpha x, \alpha y) = (\alpha(x+y), \alpha x, \alpha y)$$
$$= \alpha((x+y), x, y) = \alpha T(u).$$

Lema 2.55. *Sejam U e V espaços vetoriais sobre $\mathbb{K}$ e $T : U \to V$ uma transformação linear. Então*

1. *$T(0_U) = 0_V$, 0_U é vetor nulo de U e 0_V é vetor nulo de V.*
2. *$T(-u) = -T(u)$, para todo $u \in U$.*

Demonstração.

1. Basta vermos que $0_V + T(0_U) = T(0_U) = T(0_U + 0_U) = T(0_U) + T(0_U)$. Logo, $0_V = T(0_U)$.

2. $T(-u) = T((-1) \cdot u) = (-1) \cdot T(u) = -T(u)$.

□

Em outras palavras, o Item *1.* desse lema nos diz que uma transformação linear leva o vetor nulo de U no vetor nulo de V.

O lema a seguir decorre diretamente da definição.

Lema 2.56. *Sejam U e V espaços vetoriais sobre $\mathbb{K}$. A função $T : U \to V$ é transformação linear se, e somente se,*

$$T(\alpha u_1 + u_2) = \alpha T(u_1) + T(u_2), \text{ para todos } u_1,u_2 \in U \text{ e para todo } \alpha \in \mathbb{K}.$$

Demonstração.

$\Rightarrow$ Sendo T uma transformação linear, segue que

$$T(\alpha u_1 + u_2) = T(\alpha u_1) + T(u_2) = \alpha T(u_1) + T(u_2).$$

$\Leftarrow$ Iniciaremos mostrando que $T(u_1 + u_2) = T(u_1) + T(u_2)$. Consideremos $\alpha = 1$, unidade de $\mathbb{K}$. Desse modo, $T(1 \cdot u_1 + u_2) = 1 \cdot T(u_1) + T(u_2)$ e consequentemente, $T(u_1 + u_2) = T(u_1) + T(u_2)$. Agora, basta provar que $T(\alpha u_1) = \alpha T(u_1)$. Para isso, tome $u_2 = 0$, vetor nulo de U, e então $T(\alpha u_1 + 0) = \alpha T(u_1) + T(0)$. Isso implica que $T(\alpha u_1) = \alpha T(u_1) + T(0) = \alpha T(u_1) + 0 = \alpha T(u_1)$. Logo, T é transformação linear.

□

Lema 2.57. *Sejam U e V espaços vetoriais sobre $\mathbb{K}$ e $T : U \to V$ uma transformação linear. Então $T\left(\sum_{i=1}^{m}(\alpha_i u_i)\right) = \sum_{i=1}^{m} \alpha_i T(u_i)$, onde $\alpha_i \in \mathbb{K}$ e $u_i \in U$, $i = 1, ..., m$.*

A demonstração é imediata do lema anterior.

Embora o resultado seja imediato, nos dará embasamento necessário para o teorema que se seguirá.

Consideremos e_i o vetor de $\mathbb{K}^n$ com todas as entradas iguais ao elemento neutro de $\mathbb{K}$, exceto na i-ésima, que é igual a 1, unidade de $\mathbb{K}$. Dessa forma, definimos $\beta = \{e_1, ..., e_n\}$ uma base de $\mathbb{K}^n$. Do Lema 2.57

$$T(x_1, ..., x_n) = T(\sum_{i=1}^{n} x_i e_i) = \sum_{i=1}^{n} x_i T(e_i).$$

Em um âmbito geral, isso nos diz que o resultado da aplicação de uma transformação linear T depende da aplicação dessa transformação na base canônica de $\mathbb{K}^n$. Se sabemos como a transformação linear age sobre a base β, podemos estender para todo o espaço vetorial.

Definição 2.58. *Sejam U e V dois espaços vetoriais sobre um corpo $\mathbb{K}$ e $T : U \to V$ uma transformação linear. Definimos:*

***Núcleo** de T, denotado por $NucT$, o conjunto $\{u \in U \mid T(u) = 0\}$.*

***Imagem** de T, denotado por ImT, o conjunto $\{v \in V \mid \exists\, u \in U$ onde $T(u) = v\}$.*

Proposição 2.59. *Sejam U e V dois espaços vetoriais sobre um corpo $\mathbb{K}$ e $T : U \to V$ uma transformação linear. Então*

1. *$NucT$ é um subespaço vetorial de U e ImT é um subespaço vetorial de V.*
2. *T é injetora se, e somente se, $NucT = \{0\}$.*

Demonstração.

1. Decorre diretamente da definição de núcleo que $NucT$ é subconjunto de U. Pelo Lema 2.55, temos que $T(0_U) = 0_V$. Portanto, $0_U \in NucT$ e além disso, $NucT \neq \emptyset$. Sejam agora x,$y \in NucT$. Então $T(x + y) = T(x) + T(y) = 0 + 0 = 0$. Portanto, $x + y \in NucT$. Consideremos agora $x \in NucT$ e $\alpha \in \mathbb{K}$. $T(\alpha x) = \alpha T(x) = \alpha 0 = 0$. Portanto, $\alpha u \in NucT$. Logo, $NucT$ é subespaço vetorial de U.

Decorre diretamente da definição que ImT é subconjunto de V. Pelo Lema 2.55, temos que $T(0_U) = 0_V$. Portanto, $0_V \in ImT$ e além disso, $ImT \neq \emptyset$. Sejam agora x,$y \in ImT$. Então existem $u_1, u_2 \in U$ tais que $x = T(u_1)$ e $y = T(u_2)$. Assim $x + y = T(u_1) + T(u_2) = T(u_1 + u_2)$. Portanto, $x + y \in ImT$. Consideramos agora $x = T(u_1) \in ImT$ e $\alpha \in \mathbb{K}$. $\alpha x = \alpha T(u_1) = T(\alpha u_1)$. Portanto, $\alpha x \in ImT$. Logo, ImT é subespaço vetorial de V.

2. ($\Rightarrow$) Vamos considerar u pertencente ao núcleo de T, então $T(u) = 0_V$. Mas como T é uma transformação linear, do Item 1 do Lema 2.55 temos que $T(0_U) = 0_V$, ou seja, 0_U está no núcleo de T. Como T é injetora, temos apenas um elemento em U que é levado em 0_V, ou seja, $u = 0_U$. Logo, $NucT = \{0\}$.

($\Leftarrow$) Sejam agora $T(x) = T(y)$. Isso implica que $0 = T(x) - T(y) = T(x - y)$. Mas então temos que $x - y$ pertence ao núcleo de T. Da hipótese, $NucT = \{0\}$, ou seja, nos resta que $x - y = 0_U$ e isso ocorre se, e somente se, $x = y$. Logo, T é injetora.

□

Os resultados e definições que se deram a partir da Definição 2.49 foram apresentados, sobretudo, para que pudéssemos enunciar a proposição acima. Essa proposição nos mostra

que existem duas maneiras de se descrever subespaços vetoriais C: por meio da imagem ou do núcleo de uma transformação linear T. De forma resumida, nos importa saber que dada uma transformação linear T, temos que $Im\,T$ e $Nuc\,T$ são subespaços vetoriais bem definidos. Futuramente, a estruturação e definição dos códigos lineares serão derivados desses conceitos.

A seguir, apresentamos o último resultado deste capítulo.

Teorema 2.60. *Sejam U e V dois espaços vetoriais sobre $\mathbb{K}$ onde a dimensão de U é finita e $T : U \to V$ uma transformação linear. Então*

$$dim\,U = dim\,Nuc\,T + dim\,Im\,T.$$

Demonstração.

Como a dimensão de U é finita, temos que a dimensão do $Nuc\,T$ também é. Seja k a dimensão do $Nuc\,T$. Vamos considerar $\beta = \{u_1, ...,u_k\}$ uma base do $Nuc\,T$.

Agora, determinamos β' ao acrescentarmos elementos ao conjunto β, buscando completar a base do $Nuc\,T$ para uma base de U, com n elementos: $\beta' = \{u_1, ...,u_k, u_{k+1}, ...,u_n\}$. Vamos provar que $\{T(u_{k+1}), ..., T(u_n)\}$ gera $Im\,T$. Consideremos um elemento u qualquer pertencente a U. Então u pode ser escrito como combinação linear dos elementos de β', ou seja, $u = \alpha_1u_1 + ... + \alpha_ku_k + \alpha_{k+1}u_{k+1} + ... + \alpha_nu_n$, $\alpha_i \in \mathbb{K}$. Ao aplicarmos T em u temos que:

$$T(u) = T(\alpha_1u_1 + ... + \alpha_ku_k + \alpha_{k+1}u_{k+1} + ... + \alpha_nu_n).$$

Isso implica que

$$\begin{aligned} T(u) &= T(\alpha_1u_1) + ... + T(\alpha_ku_k) + T(\alpha_{k+1}u_{k+1}) + ... + T(\alpha_nu_n) \\ &= \alpha_1T(u_1) + ... + \alpha_kT(u_k) + \alpha_{k+1}T(u_{k+1}) + ... + \alpha_nT(u_n). \end{aligned}$$

Como $u_i, 1 \leq i \leq k$, são elementos do $Nuc\,T$, temos $T(u_i) = 0$, logo,

$$T(u) = \alpha_{k+1}T(u_{k+1}) + ... + \alpha_nT(u_n).$$

Em outras palavras, qualquer elemento da $Im\,T$ pode ser escrito como combinação linear dos elementos $\{T(u_{k+1}), ..., T(u_n)\}$.

Agora, mostraremos que eles são linearmente independentes. Vamos supor que $\alpha_{k+1}T(u_{k+1}) + ... + \alpha_nT(u_n) = 0$. Então $T(\alpha_{k+1}u_{k+1} + ... + \alpha_nu_n) = 0$. Ou seja, $(\alpha_{k+1}u_{k+1} + ... + \alpha_nu_n)$ pertence ao $Nuc\,T$. Mas como $\{u_1, \ldots, u_k\}$ é uma base de $Nuc\,T$ e os vetores $u_{k+1}, ..., u_n$ são linearmente independentes dos $u_1, \ldots, u_k$, então a única forma dessa combinação estar no núcleo é se $\alpha_i = 0$ para todo i, $k + 1 \leq i \leq n$.

Portanto, os vetores $T(u_{k+1}), \ldots, T(u_n)$ são linearmente independentes e, consequentemente eles formam uma base de $Im\,T$, o que mostra que $dim\,Im\,T = n - k$.

Logo, $dim\,Nuc\,T + dim\,Im\,T = k + (n - k) = n = dimU$.

□

Mais adiante, verificaremos que os códigos lineares determinados a partir do núcleo de uma transformação possuem certa vantagem: esses códigos demandam menor capacidade computacional para serem utilizados. Esse teorema servirá de apoio especialmente quando formos entender como se dá a característica desses códigos.

Com todos esses conceitos prévios estabelecidos, estamos prontos para iniciar efetivamente nosso estudo. No próximo capítulo, começaremos a apresentar os conceitos fundamentais para a compreensão dos códigos corretores de erros.

3 CÓDIGOS CORRETORES DE ERROS

Neste capítulo daremos uma breve introdução aos códigos corretores de erros, exibindo os conceitos básicos de códigos. As principais referências para este capítulo foram Coelho e Lourenço (2013), Hefez e Vilella (2008) e Ventura (2013).

3.1 Estrutura de um código

Vamos iniciar analisando algumas situações cotidianas. Primeiramente, um idioma é a forma mais usual de se explicar o que é necessário para estruturar um código. Considere o alfabeto da língua portuguesa $\mathcal{A}$, composto por 23 letras mais o espaço, ç e as vogais acentuadas. Temos que levar em consideração que a maior palavra da língua portuguesa é a palavra pneumoultramicroscopicossilicovulcanoconiótico, composta por 46 letras. Dessa forma, a língua portuguesa é um código de $\mathcal{A}^{46} = \underbrace{\mathcal{A} \times \mathcal{A} \times \ldots \times \mathcal{A}}_{\text{46 vezes}}$, mais ainda, a língua portuguesa é um subconjunto $\mathcal{P} \subset \mathcal{A}^{46}$.

Agora vamos supor que a transmissão de uma informação seja modificada por algum tipo de ruído. Uma pessoa recebe uma mensagem dizendo "*pinte a tarede*". Fica fácil identificar que a palavra *tarede* não pertence ao idioma, consequentemente não pertence ao conjunto de palavras da língua portuguesa $\mathcal{P}$. Dessa forma, realizamos a correção verificando que a palavra mais próxima que pertence ao idioma é a palavra *parede*.

No entanto, partindo da abordagem do estudo de códigos, não podemos considerar o código $\mathcal{P}$ como sendo muito eficiente já que existem muitas palavras próximas umas das outras. Por exemplo, vamos supor que ao transmitir a mensagem "*comprei um gato*", a mensagem que chega ao receptor é "comprei um lato". Nesse caso a correção se torna mais difícil pelo fato de existirem muitos elementos de $\mathcal{P}$ que são próximos da palavra que chegou errada. O método de correção dessa área poderia assumir qualquer um dos elementos como rato, pato, mato, bato, etc, de $\mathcal{P}$ como sendo a palavra correta. Assim, o objetivo da teoria é desenvolver métodos que permitam detectar e corrigir estes erros.

Com esse exemplo, podemos definir quais são os elementos necessários para a estruturação de um código.

- Um conjunto finito $\mathcal{A}$ que chamaremos de **alfabeto**. Usualmente trabalharemos com o alfabeto sendo o corpo $\mathbb{Z}_p = \{\overline{0}, \overline{1}, ..., \overline{p-1}\}$, p primo.
- Sequência finita de símbolos do alfabeto. Cada sequência única chamaremos de **palavra** onde todas essas palavras formam um conjunto V. O número de letras de uma palavra chama-se o seu **comprimento**.

Em nosso estudo, como estamos lidando com códigos que são efetivos, trabalharemos com conjuntos de palavras que têm o mesmo comprimento. Também, o conjunto de todas as palavras será o espaço vetorial $V = \mathcal{A}^n$.

Um exemplo de palavra é o elemento $v = (0,1,0,1,1)$ pertencente ao espaço vetorial $\mathbb{Z}_2^5$ sobre o corpo $\mathbb{Z}_2$. Note que nesse caso o alfabeto é o corpo $\mathbb{Z}_2$.

- Um **código** de comprimento n sobre um alfabeto $\mathcal{A}$ será um subconjunto qualquer C de palavras, onde $C \subseteq V = \mathcal{A}^n$. Quando a quantidade de elementos do alfabeto de um código é p, dizemos que o código é p-ário.

Exemplo 3.1. *Um exemplo de código é o conjunto* $C = \{(0,0,0,0,0), (0,1,0,1,1), (1,0,1,1,0), (1,1,1,0,1)\} \subset V = \mathbb{Z}_2^5$*, onde o alfabeto é o corpo* $\mathbb{Z}_2$*. Neste caso, dizemos que* C *é um código binário, com quatro palavras e de comprimento 5.*

Apresentaremos agora quais os elementos de um sistema de comunicação.

- Fonte: local onde a mensagem original é produzida;
- Codificador da fonte: local onde a mensagem original é transformada em código de fonte, ou seja, onde é codificada.
- Codificador de canal: ocorre uma recodificação, do código fonte em código de canal, onde são acrescentadas informações. Essa codificação é que possibilita a correção de possíveis erros.
- Canal: meio de transmissão que o código será enviado. No canal ocorrem ruídos que podem modificar a mensagem original.
- Decodificador de canal: recodificação do código de canal em código fonte, novamente.
- Decodificador da fonte: transformação do código da fonte na mensagem original.
- Destinatário: receptor da mensagem.

Para melhor ilustrar o papel de cada elemento, vamos analisar o exemplo do código de um robô. Vamos supor que temos um robô centrado em um tabuleiro de xadrez onde podemos dar quatro comandos para esse robô: caminhar uma casa para o norte, caminhar uma casa para o sul, caminhar uma casa para o leste e caminhar uma casa para o oeste. Esses comandos são as mensagens originais. Podemos codificar as mensagens originais como elementos de $\mathbb{Z}_2^2$. Esses elementos são nossos códigos da fonte, que registramos na Tabela 1 a seguir:

Tabela 1 – Códigos da fonte

Fonte	Código da fonte
Leste	00
Oeste	01
Norte	10
Sul	11

Fonte: Autor.

Agora vamos recodificar o código da fonte o transformando em código de canal. Fazemos isso acrescentando informações a ele. Essas informações são chamadas de redundâncias. A forma como as redundâncias são introduzidas será explicada no próximo capítulo. Por hora, apenas faremos o acréscimo, que está registrada na Tabela 2 abaixo.

Tabela 2 – Códigos de canal

Código da fonte	Código da canal
00	00000
01	01011
10	10110
11	11101

Fonte: Autor.

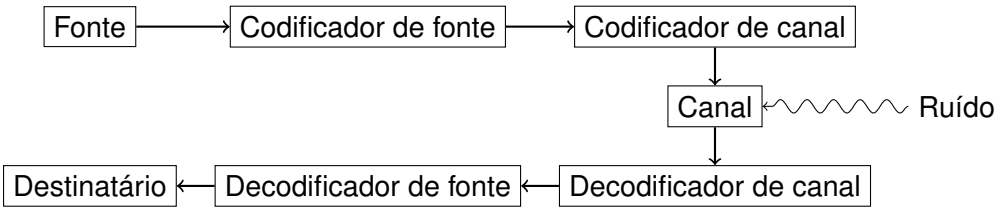
Esses elementos formam o conjunto do Exemplo 3.1
$C = \{(0,0,0,0,0), (0,1,0,1,1),(1,0,1,1,0),(1,1,1,0,1)\} \subset \mathbb{Z}_2^5$.

Após tudo isso, o código é transmitido por um canal de comunicação. Ao ser transmitido pelo canal, tem a possibilidade dos ruídos modificarem a mensagem original. Vamos supor que uma mensagem v seja enviada, o decodificador de canal recebe a mensagem $v' = (1,0,0,0,0)$. Ao compararmos com os elementos do conjunto C, fica evidente que essa mensagem não pertence a esse conjunto, ou seja, nós conseguimos constatar que a mensagem foi modificada pelos ruídos.

No entanto, de grosso modo, podemos identificar que o elemento $(0,0,0,0,0)$ de C é o “mais parecido"com v'. Dessa forma definimos que $v = (0,0,0,0,0)$. Por fim, a mensagem seria decodificada da fonte e a mensagem chegaria fielmente ao usuário. Posteriormente mostraremos matematicamente como isso ocorre, recorrendo aos conceitos definidos nessa área de estudo.

A Figura 1 abaixo ilustra o processo de codificação e decodificação.

Figura 1 – Processo de codificação e decodificação



Fonte: Autor.

Vamos supor agora que enviamos a mesma mensagem antes de codificar como código de canal, ou seja, ela foi enviada como código da fonte. Quando enviada a mensagem $v = (0,0)$ e ela ser modificada por ruídos, vamos supor que o decodificador da fonte recebe a mensagem $v' = (1,0)$. Nesse caso o erro não pode ser detectado, pois o elemento $(1,0)$ também pertence ao conjunto de mensagens. Ao realizar a decodificação da fonte, o usuário receberia a mensagem “Norte"ao invés de “Leste". Sendo assim, a impossibilidade de correção faz com que o receptor adquira uma informação diferente da que foi enviada.

Na seção seguinte, estudaremos os conceitos iniciais necessários para o estudo de códigos.

3.2 Métrica de Hamming

Tratando-se de notações, utilizaremos $\mathcal{A}^n$ para nos referirmos ao conjunto de palavras de comprimento n sobre um alfabeto $\mathcal{A}$ e C para nos referirmos à um subconjunto de palavras de $\mathcal{A}^n$, quando esse subconjunto for um código. Referindo-se à quantidade de elementos de um conjunto V, empregaremos a notação $|V|$.

Definição 3.2. *Dados dois elementos* $u = (u_1,u_2,...,u_n)$, $v = (v_1,v_2,...,v_n) \in \mathcal{A}^n$ *e considerando os pares ordenados* (u_i, v_i); $1 \leq i \leq n$, *a* ***Distância de Hamming*** *de* u *e* v *é definida por*

$$d(u,v) = |D| \text{ onde } D = \{(u_i, v_i) \,|\, u_i \neq v_i, 1 \leq i \leq n\}.$$

Essa definição nos diz que a distância de duas palavras é designada através da quantidade de elementos de um conjunto, onde esse conjunto é formado por pares ordenados. Para cada par ordenado, a abscissa é o elemento u_i e a ordenada é o elemento v_i de modo que $u_i \neq v_i$, i variando de 1 até n.

Exemplo 3.3. *Seja* $C = \{(0,0,0,0,0), (0,1,0,1,1),(1,0,1,1,0),(1,1,1,0,1)\} \subset \mathbb{Z}_2^5$ *um código. As distâncias de Hamming dos elementos deste código são:*

$$\begin{aligned}
&d(\,(0,0,0,0,0),(0,1,0,1,1)\,) = 3 = |\{(u_2,v_2), (u_4,v_4), (u_5,v_5)\}|;\\
&d(\,(0,0,0,0,0),(1,0,1,1,0)\,) = 3 = |\{(u_1,v_1), (u_3,v_3), (u_4,v_4)\}|;\\
&d(\,(0,0,0,0,0),(1,1,1,0,1)\,) = 4 = |\{(u_1,v_1), (u_2,v_2), (u_3,v_3), (u_5,v_5)\}|;\\
&d(\,(0,1,0,1,1),(1,0,1,1,0)\,) = 4 = |\{(u_1,v_1), (u_2,v_2), (u_3,v_3), (u_5,v_5)\}|;\\
&d(\,(0,1,0,1,1),(1,1,1,0,1)\,) = 3 = |\{(u_1,v_1), (u_3,v_3), (u_4,v_4)\}|;\\
&d(\,(1,0,1,1,0),(1,1,1,0,1)\,) = 3 = |\{(u_2,v_2), (u_4,v_4), (u_5,v_5)\}|.
\end{aligned}$$

Proposição 3.4. *Dados* u,v *e* $w \in \mathcal{A}^n$, *as seguintes propriedades são válidas.*

1. *Positividade:* $d(u,v) \geq 0$, *valendo a igualdade quando* $u = v$.
2. *Simetria:* $d(u,v) = d(v,u)$.
3. *Desigualdade Triangular:* $d(u,v) \leq d(u,w) + d(w,v)$.

Demonstração.

1. Seja $u = (u_1, ..., u_n)$ e $v = (v_1, ..., v_n)$.

Vamos considerar $u \neq v$, ou seja, existe ao menos um par de coordenadas pertecente ao conjunto D onde $D = \{(u_i, v_i) \,|\, u_i \neq v_i, 1 \leq i \leq n\}$. Portanto, $d(u,v) > 0$.

Verifiquemos o caso em que $u = v$. O conjunto de entradas, coordenada a coordenada, $D = \{(u_i, v_i) \,|\, u_i \neq v_i, 1 \leq i \leq n\} = \emptyset$. Portanto, $d(u,v) = 0$.

Logo, $d(u,v) \geq 0$.

Agora seja $d(u,v) = 0$, isso implica que $D = \{(u_i, v_i) \,|\, u_i \neq v_i, 1 \leq i \leq n\} = \emptyset$ e portanto, $|D| = 0$. Mas isso significa que $u_i = v_i$ para todo u_i,v_i, $1 \leq i \leq n$. Logo, $u = v$.

2. Sejam $u = (u_1, ..., u_n)$ e $v = (v_1, ..., v_n)$. Temos $d(u,v) = |D|$, onde $D = \{(u_i, v_i) \,|\, u_i \neq v_i, 1 \leq i \leq n\} = \{(v_i, u_i) \,|\, v_i \neq u_i, 1 \leq i \leq n\}$, assim, $d(v,u) = |D|$. Logo, $d(u,v) = d(v,u)$.

3. Dados $u = (u_1, ..., u_n)$ e $v = (v_1, ..., v_n)$, temos dois casos: $u_i = v_i$ para todo i e $u_i \neq v_i$, para algum $1 \leq i \leq n$.

No caso em que $u_i = v_i$ para todo i, esses pares ordenados não acrescentam nada na soma em $d(u,v)$, logo, é evidente que $0 = d(u,v) \leq d(u,w) + d(w,v)$ para todo $w \in A^n$.

Agora, no caso em que $u_i \neq v_i$ para algum $1 \leq i \leq n$, esses pares ordenados somam 1 em $d(u,v)$. Como não podemos ter $u_i = w_i$ e $w_i = v_i$ ocorrendo ao mesmo tempo, temos que os pares ordenados (u_i,w_i) e (w_i,v_i) somam 1 ou 2 em $d(u,w) + d(w,v)$. Portanto, $d(u,v) \leq d(u,w) + d(w,v)$.

□

Notemos que, particularmente em nossos estudos, como os códigos pertencem a espaços vetoriais constituídos sobre o corpo $\mathbb{K} = \mathbb{Z}_2$, temos apenas duas possibilidades para u_i,v_i e w_i, que são 0 ou 1. Não podendo assumir $u_i = w_i$ e $w_i = v_i$ ocorrendo ao mesmo tempo significa que em $d(u,w) + d(w,v)$ somam exatamente 1, para cada i nesta situação. Ou seja, em nossos estudos, essa desigualdade é estabelecida estritamente no caso em que $u_i = v_i$, pois, consequentemente $w_i = u_i = v_i$ vai somar 0 em $d(u,w) + d(w,v)$ (caso da igualdade) ou $w_i \neq u_i = v_i$ vai somar 1 na mesma. Assim teremos a desigualdade triangular.

Definição 3.5. *Dados um elemento $u \in \mathcal{A}^n$ e um número natural $t \geq 0$, definimos:*

***Disco de centro em** u **e raio** t, o conjunto*

$$D(u,t) = \{v \in \mathcal{A}^n \,|\, d(u,v) \leq t\}.$$

***Esfera de centro em** u **e raio** t, o conjunto*

$$S(u,t) = \{v \in \mathcal{A}^n \,|\, d(u,v) = t\}.$$

Lema 3.6. *Para todo $u \in \mathcal{A}^n$ e todo número natural $t > 0$ temos que*

$$|D(u,t)| = \sum_{i=0}^{t} \binom{n}{i} (q-1)^i$$

onde n é o comprimento das palavras e q a quantidade de elementos do alfabeto $\mathcal{A}$.

Demonstração. A quantidade de elementos do disco centrado em u e de raio t é dado pela soma de todas as esferas centradas em u e raio i, com i variando de 0 até t. Ou seja,

$$|D(u,t)| = \sum_{i=0}^{t} |S(u,i)|.$$

Mostraremos quantos elementos tem em $S(u,i)$.

Sejam $S(u,i)$ e x pertencente a $S(u,i)$. É vidente que $d(u,x) = i$. Melhor dizendo, partindo da palavra u para chegar na palavra x, é necessária a troca exata de i coordenadas. Existem $\binom{n}{i}$ formas de determinar quais dessas coordenadas serão trocadas. Agora, em cada posição alterada, teremos a substituição da letra inicial por outra letra do alfabeto A. Para isso, temos $q-1$ alternativas. Como são i mudanças, temos $(q-1)^i$ possibidades. Portanto, $|S(u,i)| = \binom{n}{i}.(q-1)^i$. Logo, $|D(u,t)| = \sum_{i=0}^{t} \binom{n}{i}.(q-1)^i$.

□

Definição 3.7. *Seja C um código. Chamamos de distância mínima de C o número natural d, onde*

$$d = min\{d(u,v) \,|\, u,v \in C, u \neq v\}.$$

No caso do Exemplo 3.3, a distância mínima de C é igual a 3. A forma como fizemos nesse exemplo, onde determinamos a distância do código C partindo do cálculo da distância de seus elementos, dois a dois, é um método utilizado para códigos com uma cardinalidade considerada pequena, pois ficaria inviável realizar o mesmo procedimento para um código com 30 elementos, por exemplo, já que seriam necessários o cálculo de $\binom{30}{2}$ casos.

Definição 3.8. *Dado um código C com distância mínima d, definimos*

$$\kappa = \left[\frac{d-1}{2}\right],$$

onde $[t]$ é a parte inteira de um número real t.

Lema 3.9. *Seja C um código com distância mínima d. Se u e v são palavras distintas de C, então*

$$D(u,\kappa) \cap D(v,\kappa) = \emptyset.$$

Demonstração.

Vamos supor, por absurdo, que existe x pertencente a C, de modo que $x \in D(u,\kappa) \cap D(v,\kappa)$. Da desigualdade triangular temos que $d(u,v) \leq d(u,x) + d(x,v)$. Além disso, como x pertence ao $D(u,\kappa)$ e x pertente ao $D(v,\kappa)$, então $d(u,x) \leq \kappa$ e $d(v,x) \leq \kappa$ e consequentemente $d(u,v) \leq d(u,x)+d(x,v) \leq \kappa+\kappa$. Da definição de κ, temos que $\kappa \leq \frac{d-1}{2}$, então,

$$d(u,v) \leq d(u,x) + (x,v) \leq \kappa + \kappa \leq \frac{d-1}{2} + \frac{d-1}{2} = d-1 < d.$$

Uma contradição, pois a distância mínima é d. Logo, $D(u,\kappa) \cap D(v,\kappa) = \emptyset$

□

Teorema 3.10. *Seja C um código com distância mínima d. Então C pode corrigir até $\kappa = \left[\frac{d-1}{2}\right]$ erros e detectar $d-1$ erros.*

Demonstração.

Vamos considerar a palavra c pertencente ao código C. Se ao transmitirmos a palavra c recebermos a palavra r com t erros, $t \leq \kappa$, temos $d(c,r) = t \leq \kappa$. Assim, $r \in D(c,\kappa)$. Pelo Lema 3.9 concluímos que $t = d(c,r) < d(c',r)$ qualquer que seja c', ou seja, a palavra mais próxima de r que pertence ao código C é c. Sendo assim, estabelecemos c a partir de r. Por outro lado, ao transmitirmos uma palavra c, podem ocorrer $d-1$ erros nela sem que ela seja transformada em outra palavra do código. Dessa forma é possível detectar a presença de erros na transmissão.

□

O código do Exemplo 3.3, é possível corrigir 1 erro, já que $1 = \left[\frac{3-1}{2}\right]$ e detectar $3-1=2$ erros.

Definição 3.11. *Seja $C \subset \mathcal{A}^n$ um código com distância mínima d e seja $\kappa = \left[\frac{d-1}{2}\right]$. O código C será dito **perfeito** se*

$$\bigcup_{v \in C} D(v,\kappa) = \mathcal{A}^n.$$

O Teorema 3.10 nos apresenta um modo para corrigir possíveis erros de uma palavra transmitida. Seja C um código com distância mínima d e $\kappa = \left[\frac{d-1}{2}\right]$ a quantidade de erros que podem ser corrigidos pelo código. Ao enviarmos uma palavra u e recebermos a palavra r, modificada, podem ocorrer dois casos:

1. A palavra r está em $D(u,\kappa)$. Nesse caso, a quantidade de erros é menor que κ. Assim, apenas substituimos r por u.

2. A mensagem r não está em $D(u,\kappa)$.

Para este último caso, restam duas opções: r está em $D(v,\kappa),\ v \neq u$ ou r não está em nenhum círculo de raio κ, centrado em algum elemento de C. No primeiro caso, a correção seria feita de maneira equivocada, já que a quantidade de erros maior que κ fez com que a palavra original se aproximasse da palavra $v \neq u$. No segundo caso a correção se torna inviável, já que a palavra r não está próxima de nenhuma palavra de C.

São três parâmetros que devemos considerar quando falamos de estudo de códigos: a quantidade de elementos M de um código C, o comprimento n dos elementos desse código e a distância d de C.

Tendo isso em mente, busca-se estruturar códigos com melhor eficácia. Para isso, essa estruturação se consiste em criar códigos com d e M relativamente grandes quando comparados à n, pois, quanto maior essa diferença, maior a possibilidade de recairmos no caso 1.

3.3 Códigos Equivalentes

Definidos os códigos corretores de erros, busca-se definir a noção de equivalência desses elementos, assim como também é feito em outras áreas da matemática. Essa noção pode ser feita a partir de definição de isometria, apresentada a seguir.

Definição 3.12. *Seja $\mathcal{A}$ um alfabeto e n um número natural. Diremos que uma função $F : \mathcal{A}^n \longrightarrow \mathcal{A}^n$ é uma isometria de $\mathcal{A}^n$ se essa aplicação preserva a distância de Hamming, ou seja,*

$$d(F(u),F(v)) = d(u,v);\ \forall u,v \in \mathcal{A}^n.$$

Proposição 3.13. *Toda isometria de $\mathcal{A}^n$ é uma bijeção.*
Demonstração.

Seja $F : \mathcal{A}^n \longrightarrow \mathcal{A}^n$ uma isometria. Vamos supor que para $u,v \in \mathcal{A}^n$ temos que $F(u) = F(v)$. Sendo assim, $d(F(u),F(v)) = d(u,v) = 0$, implicando que $u = v$. Logo, F é injetora. Sendo uma aplicação injetora de $\mathcal{A}^n$ em $\mathcal{A}^n$, temos que a imagem de F é o conjunto $\mathcal{A}^n$. Consequentemente, F é sobrejetora. Logo, F é bijetora.

□

Proposição 3.14. *Valem as seguintes propriedades:*

1. *A função identidade de $\mathcal{A}^n$ é uma isometria.*
2. *Se F é uma isometria de $\mathcal{A}^n$, então F^{-1} é uma isometria de $\mathcal{A}^n$.*
3. *Se F,G são isometrias de $\mathcal{A}^n$, então $F \circ G$ é uma isometria de $\mathcal{A}^n$.*

Por fim, vamos definir quando dois códigos são equivalentes.

Definição 3.15. *Dados C e C' dois códigos sobre $\mathcal{A}^n$, diremos que C e C' são códigos equivalentes se existir uma isometria F de $\mathcal{A}^n$ de modo que $F(C) = C'$.*

Tratando-se de equivalência e levando em consideração essa definição e os últimos dois resultados, ressaltamos algumas informações.

1. Do item 1 da Proposição 3.14, temos que vale a propriedade reflexiva.
2. Do item 2 da Proposição 3.14, temos que vale a propriedade simétrica.
3. Do item 3 da Proposição 3.14, temos que vale a propriedade transitiva.

Ou seja, podemos dizer que a isometria é uma relação de equivalência.

Em seguida, traremos exemplos de duas isometrias bem conhecidas no estudo de código.

Proposição 3.16. *Se $f : \mathcal{A} \longrightarrow \mathcal{A}$ é uma bijeção e i é um inteiro, $1 \leq i \leq n$, então a aplicação*

$$\begin{aligned} T_f^i : \mathcal{A}^n &\longrightarrow \mathcal{A}^n \\ (a_1, \ldots, a_n) &\longmapsto (a_1, \ldots, f(a_i), \ldots, a_n) \end{aligned}$$

é uma isometria.

Demonstração.

Sejam $(a_1,...,a_i,...,a_n)$ e $(b_1,...,b_i,...,b_n) \in \mathcal{A}^n, 1 \leq i \leq n$.

Como f é bijetora, $f(a_i) = f(b_i)$ se, e somente se $a_i = b_i$ para todo $a_i,b_i \in \mathcal{A}^n$. Ao aplicarmos T_f^i em $(a_1,...,a_i,...,a_n)$ e $(b_1,...,b_i,...,b_n)$ recaíremos em dois casos a seres analisados: $f(a_i) = f(b_i),$ somam 0 na distância de Hamming e $f(a_i) \neq f(b_i)$ somam 1 na distância de Hamming.

No caso em que $f(a_i) = f(b_i)$ implica que $a_i = b_i$. Isso significa que o valor na contribuição da distâcia de Hamming da i-ésima coordenada se mantém 0. No caso em que $f(a_i) \neq f(b_i)$ implica que $a_i \neq b_i$. Isso significa que o valor na contribuição da distância de Hamming da i-ésima coordenada se mantém 1. Ou seja, em ambos casos a distância de Hamming é preservada. Logo, T_f^i é uma isometria.

□

Essa Proposição nos diz que ao aplicarmos f na i-ésima coordenada de todos elementos de A^n, a distância de Hamming será preservada desde que f seja bijetora. Veremos isso no caso específico abaixo.

Avaliaremos $C = \{(0,0,0,0,0), (0,1,0,1,1),(1,0,1,1,0),(1,1,1,0,1)\} \subset \mathbb{Z}_2^5$.

Anteriormente, ao realizarmos as possíveis combinações para cálculos da distância de Hamming desse código, obtivemos os seguintes resultados:

$$\begin{aligned} &d(\,(0,0,0,0,0),(0,1,0,1,1)\,) = 3 = |\{(u_2,v_2),(u_4,v_4),(u_5,v_5)\}|; \\ &d(\,(0,0,0,0,0),(1,0,1,1,0)\,) = 3 = |\{(u_1,v_1),(u_3,v_3),(u_4,v_4)\}|; \\ &d(\,(0,0,0,0,0),(1,1,1,0,1)\,) = 4 = |\{(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_5,v_5)\}|; \\ &d(\,(0,1,0,1,1),(1,0,1,1,0)\,) = 4 = |\{(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_5,v_5)\}|; \\ &d(\,(0,1,0,1,1),(1,1,1,0,1)\,) = 3 = |\{(u_1,v_1),(u_3,v_3),(u_4,v_4)\}|; \\ &d(\,(1,0,1,1,0),(1,1,1,0,1)\,) = 3 = |\{(u_2,v_2),(u_4,v_4),(u_5,v_5)\}|. \end{aligned}$$

Consideremos agora a função bijetora

$$\begin{aligned} f : \mathbb{Z}_2 &\longrightarrow \mathbb{Z}_2 \\ a &\longmapsto a+1 \end{aligned}$$

Ao aplicarmos f na segunda coordenada de todos elementos de C, isto é, aplicando T_f^2 teremos o seguinte código:

$$C' = \{(0,1,0,0,0), (0,0,0,1,1),(1,1,1,1,0),(1,0,1,0,1)\} \subset \mathbb{Z}_2^5.$$

Agora vamos calcular a distância de Hamming dois a dois, de cada um dos elementos do código C'.

$$d(\,(0,1,0,0,0),(0,0,0,1,1)\,) = 3 = |\{(u_2,v_2),(u_4,v_4),(u_5,v_5)\}|;$$
$$d(\,(0,1,0,0,0),(1,1,1,1,0)\,) = 3 = |\{(u_1,v_1),(u_3,v_3),(u_4,v_4)\}|;$$
$$d(\,(0,1,0,0,0),(1,0,1,0,1)\,) = 4 = |\{(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_5,v_5)\}|;$$
$$d(\,(0,0,0,1,1),(1,1,1,1,0)\,) = 4 = |\{(u_1,v_1),(u_2,v_2),(u_3,v_3),(u_5,v_5)\}|;$$
$$d(\,(0,0,0,1,1),(1,0,1,0,1)\,) = 3 = |\{(u_1,v_1),(u_3,v_3),(u_4,v_4)\}|;$$
$$d(\,(1,1,1,1,0),(1,0,1,0,1)\,) = 3 = |\{(u_2,v_2),(u_4,v_4),(u_5,v_5)\}|.$$

Note que, no código C' além da distância de Hamming ser preservada, ela continua sendo determinada pelas mesmas i-ésimas coordenadas dos elementos do código C.

Proposição 3.17. *Se π é uma bijeção do conjunto $\{1,...,n\}$ nele próprio, também chamada de permutação de $\{1,...,n\}$, a aplicação permutação de coordenadas*

$$\begin{array}{rccc} T_\pi\colon & \mathcal{A}^n & \longrightarrow & \mathcal{A}^n \\ & (a_1,...,a_n) & \longmapsto & (a_{\pi(1)},...,a_{\pi(n)}) \end{array}$$

é uma isometria.

Demonstração.

Sejam $(a_1,...,a_i,...,a_n),\ (b_1,...,b_i,...,b_n) \in \mathcal{A}^n, 1 \leq i \leq n$.

Como π é bijetora, temos que $\pi(i) = \pi(j)$ se, e somente se $i = j$ para todo $i,j \in \{1,...,n\}$. Então, ao aplicarmos T_π em $(a_1,...,a_i,...,a_n)$ e $(b_1,...,b_i,...,b_n)$ recaíremos em dois casos a serem analisados, para cada coordenada: $a_{\pi(i)} = b_{\pi(i)}$ e $a_{\pi(i)} \neq b_{\pi(i)}$.

No caso em que $a_{\pi(i)} = b_{\pi(i)}$ implica que $a_i = b_i$. Isso significa que o valor na contribuição da distâcia de Hamming da i-ésima coordenada se mantém 0. No caso em que $a_{\pi(i)} \neq b_{\pi(i)}$ implica que $a_i \neq b_i$. Isso significa que o valor na contribuição da distância de Hamming da i-ésima coordenada se mantém 1. Ou seja, em ambos casos a distância de Hamming é preservada. Logo, T_π é uma isometria.

□

Teorema 3.18. *Seja $F : \mathcal{A}^n \longrightarrow \mathcal{A}^n$ uma isometria, então existem uma permutação π de $\{1,...,n\}$ e bijeções f_i de $\mathcal{A}, i = 1,...,n$ tais que*

$$F = T_\pi \circ T^1_{f_1} \circ ... \circ T^n_{f_n}.$$

A demonstração deste teorema pode ser encontrada em (Hefez e Villela, 2008).

Decorrente da definição de códigos equivalentes e do teorema apresentado, enunciamos o próximo corolário.

Corolário 3.19. *Sejam C e C' dois códigos em $\mathcal{A}^n$. Temos que C e C' são equivalentes se, e somente se, existem uma permutação π de $\{1,...n\}$ e bijeções $f_1,...,f_n$ de $\mathcal{A}$ tais que*

$$C' = \{(f_{\pi_{(1)}}(x_{\pi_{(1)}}), ..., f_{\pi_{(n)}}(x_{\pi_{(n)}})) \mid (x_1, ..., x_n) \in C\}.$$

De forma geral, dois códigos de mesmo comprimento, sobre um alfabeto $\mathcal{A}$ serão equivalentes se, e somente se, podemos obter um partindo do outro tomando como intermediação as duas seguintes operações:

1. Através de uma bijeção em $\mathcal{A}$, trocar todas as i-ésimas coordenadas em todas palavras do código;

2. Através de uma permutação em $\{1, .., n\}$, trocar todas coordenadas em todas as palavras do código.

4 CÓDIGOS LINEARES

Neste capítulo daremos início ao estudo de um tipo específico de códigos corretores de erros, os códigos lineares. Na prática, essa modalidade de código é a mais utilizada. As principais referências utilizadas são Hefez e Vilella (2008) e Ventura (2013).

4.1 Códigos Lineares

Ao longo deste capítulo, $\mathbb{K}$ denotará um corpo finito com q elementos. Assim temos $\mathbb{K}^n$ como sendo um $\mathbb{K}$-espaço vetorial de dimensão n, onde n é um número natural.

Definição 4.1. *Um código $C \subset \mathbb{K}^n$ será chamado de **código linear** se for um subespaço vetorial de $\mathbb{K}^n$.*

O código $C = \{(0,0,0,0,0), (0,1,0,1,1),(1,0,1,1,0),(1,1,1,0,1)\} \subset \mathbb{Z}_2^5$ é um subespaço vetorial. Por definição, temos que todo código linear é um espaço vetorial de dimensão finita.

Seja n a dimensão de um código C qualquer sobre um corpo $\mathbb{K}$, $\mathbb{K}$ com q elementos, e seja $\beta = \{\beta_1, ..., \beta_n\}$ uma base. Logo, todo elemento de C pode ser escrito de forma única como combinação linear dos elementos da base:

$$\alpha_1 \cdot \beta_1 + \alpha_2 \cdot \beta_2 + ... + \alpha_n \cdot \beta_n,\ \alpha_i \in \mathbb{K}, 1 \leq i \leq n.$$

Segue então que a quantidade de elementos em C será M onde o mesmo será dado por

$$M = |C| = q^n,$$

e daí, temos a relação

$$dim\ C = n = log_q M.$$

Definição 4.2. *Dado $x \in K^n$, chamamos de **peso** de $x = (x_1, ..., x_n)$ como sendo o inteiro*

$$\omega(x) = |\{x_i \,|\, x_i \neq 0, 1 \leq i \leq n, i \in \mathbb{N}\}|.$$

Em outras palavras, o peso de um elemento x é dado pela distância de Hamming desse elemento até o elemento neutro.

$$\omega(x) = d(x, 0).$$

Definição 4.3. *O peso de um código linear C é o número inteiro*

$$\omega(C) = min\{\omega(x) \,|\, x \in C\backslash\{0\}\}, \textit{ se } C \neq \{0\},\ \textit{ e } \omega(C) = 0 \textit{ se } C = \{0\}.$$

Em resumo, a distância de Hamming do elemento mais próximo ao elemento neutro, em C, determinará o peso do código C.

Proposição 4.4. *Seja $C \subset \mathbb{K}^n$ um código linear com distância mínima d. Temos que*

1. $d(x,y) = \omega(x-y), \forall x,y \in \mathbb{K}^n$.

2. $d = \omega(C)$.

Demonstração.

1. Sejam $x,y \in C, x \neq y$. Como C é um código linear, temos que $x - y \in C\backslash\{0\}$ e da definição da distância de Hamming temos $d(x,y) = |\{(x_i, y_i) \,|\, x_i \neq y_i, 1 \leq i \leq n\}|$. Da definição de peso, $\omega(x-y) = |\{x_i - y_i \neq 0, 1 \leq i \leq n, i \in \mathbb{N}\}|$. Mas note que

$$|\{x_i - y_i \neq 0, 1 \leq i \leq n, i \in \mathbb{N}\}| = |\{x_i \neq y_i, 1 \leq i \leq n, i \in \mathbb{N}\}| = d(x,y).$$

Logo, $\omega(x-y) = d(x,y)$.

2. Assumindo $C \neq \{0\}$, temos que C contém pelo menos dois elementos. Da definição de distância mínima temos $d = min\{d(x,y); x,y \in C, x \neq y\}$. Sejam então x e y tais que $d = d(x,y)$. Do item *1* temos que $d = d(x,y) = \omega(x-y)$.

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

Seja agora $x \in C$ tal que $\omega(x) = \omega(C)$, isso significa que $\omega(C) = \omega(x) = d(x,0)$.

Mas $d \leq d(x,0) = \omega(C)$, logo, $d = \omega(C)$.

□

Vale citar que, com a proposição acima passaremos a denominar distância mínima de um código como também sendo o peso do mesmo. Outro ponto importante é o fato de nós termos obtido uma otimização quanto ao cálculo da distância mínima d de um código. Seja um código C com M elementos. Se antes se faziam necessários $\binom{M}{2}$ cálculos para determinar d, agora com $M-1$ cálculos nós a obtemos, que seria o peso de todos os elementos do código, exceto o nulo.

Por mais que esse método seja mais factível do que o método apresentado inicialmente, ainda seria inviável seu uso no cálculo de códigos com um M "grande" já que demandaria um custo computacional alto. Sendo assim, mais a frente em nosso trabalho, apresentaremos métodos que exijam menor consumo computacional para determinar d.

Agora, falaremos sobre duas formas para determinar subespaços vetoriais por meio de transformações lineares. A primeira é através da imagem de uma transformação linear e a segunda é através do núcleo de uma transformação linear. Por exemplo, sejam U e V espaços vetoriais sob um mesmo corpo $\mathbb{K}$ e seja T uma transformação linear de U em V. No Capítulo 1 provamos que a imagem de T é um subespaço vetorial de V e também que o núcleo de T é um subespaço vetorial de U.

Aqui em nosso estudo sobre códigos, utilizaremos essa mesma ferramenta para determinar códigos lineares, pois pela definição de transformação linear, as operações em U e V são preservadas tanto no núcleo dessa transformação quanto na imagem, satisfazendo a definição de códigos lineares.

Vejamos o código $C = \{(0,0,0,0,0), (0,1,0,1,1),(1,0,1,1,0),(1,1,1,0,1)\}$ que já foi visto anteriormente. Assumindo $A = \mathbb{Z}_2$ como alfabeto e o código como subespaço vetorial de $\mathbb{Z}_2^5$, note que o mesmo é determinado pela imagem da transformação linear

$$\begin{array}{rccc} T: & \mathbb{Z}_2^2 & \longrightarrow & \mathbb{Z}_2^5 \\ & (x_1,x_2) & \longmapsto & (x_1,x_2,x_1,x_1+x_2,x_2) \end{array}$$

Notemos que $\mathbb{Z}_2^2 = \{(0,0), (0,1), (1,0), (1,1)\}$ tem poucos elementos, portanto, conseguimos fazer essa verificação com facilidade, aplicando T cada um desses elementos. Assim, temos

$$T(0,0) = (0,0,0,0+0,0) = (0,0,0,0,0);$$

$$T(0,1) = (0,1,0,0+1,1) = (0,1,0,1,1);$$

$$T(1,0) = (1,0,1,1+0,0) = (1,0,1,1,0);$$

$$T(1,1) = (1,1,1,1+1,1) = (1,1,1,0,1),$$

que é justamente o código C.

Aqui fica esclarecida a “regra" utilizada para acrescentar as redundâncias mencionadas no capítulo anterior.

Agora, em um âmbito geral, vamos analisar como obter um código como imagem de uma tranformação linear. Determinemos uma base $\{v_1,...,v_k\}$ do código C em um espaço vetorial $V = \mathbb{K}^n$ de dimensão n e seja $U = \mathbb{K}^k$ um espaço vetorial de dimensão k. Vamos levar em consideração a transformação

$$\begin{array}{rccc} T: & U & \longrightarrow & V \\ & x = (x_1,...,x_k) & \longmapsto & x_1v_1 + x_2v_2 + ... + x_kv_k \end{array}$$

Vejamos que, dados $x,y \in U$ com $T(x) = T(y)$, temos que

$$x_1v_1 + ... + x_kv_k = y_1v_1 + ... + y_kv_k.$$

Isso significa que $(x_1-y_1)v_1+...+(x_k-y_k)v_k = 0$. Sendo $\{v_1,...,v_n\}$ linearmente independente, a igualdade será satisfeita se, e somente se $x_i - y_i = 0$. Isso implica que $x_i = y_i, 1 \leq i \leq k$. Com isso, temos a injetividade na transformação T. Além disso, a imagem da transformação é o código C.

Exemplo 4.5. *Vamos considerar $C \subset \mathbb{K}^5$ sobre $\mathbb{Z}_2$, com base*

$\{(1,0,0,0,1),(0,1,0,0,0),(0,0,1,0,1),(0,0,0,1,1)\}$.

Queremos transmitir as palavras $x_1 = (0,0,0,1)$,$x_2 = (0,0,1,1)$,$x_3 = (1,1,0,0)$, $x_4 = (0,1,1,0)$ *pertencentes ao código da fonte em* $\mathbb{Z}_2^4$.

Então

$$T(0,0,0,1) = 0(1,0,0,0,1) + 0(0,1,0,0,0) + 0(0,0,1,0,1) + 1(0,0,0,1,1) = (0,0,0,1,1);$$
$$T(0,0,1,1) = 0(1,0,0,0,1) + 0(0,1,0,0,0) + 1(0,0,1,0,1) + 1(0,0,0,1,1) = (0,0,1,1,0);$$
$$T(1,1,0,0) = 1(1,0,0,0,1) + 1(0,1,0,0,0) + 0(0,0,1,0,1) + 0(0,0,0,1,1) = (1,1,0,0,1);$$
$$T(0,1,1,0) = 0(1,0,0,0,1) + 1(0,1,0,0,0) + 1(0,0,1,0,1) + 0(0,0,0,1,1) = (0,1,1,0,1).$$

$T(x_1)$,$T(x_2)$,$T(x_3)$,$T(x_4)$ estão na imagem da trasformação T. Se aplicarmos T em todos elementos do código-fonte, obtemos como imagem exatamente o código C, ou seja, $C = Im\,T$.

Portanto, um código linear C de dimensão k pode ser determinado por uma transformação linear de $\mathbb{K}^k$ em $\mathbb{K}^n$, cuja imagem é C.

Temos certa facilidade em determinar todos os elementos de C através desse método: basta definir uma base em C e uma transformação de $\mathbb{K}^k$ em $\mathbb{K}^n$. Mas digamos que o interesse fosse o oposto: dado um elemento v em $\mathbb{K}^n$ queremos verificar se esse elemento pertence ao subespaço C. Para isso, teríamos que aplicar T em todos os q^k elementos do código fonte, ou resolvermos o sistema de n equações com k incógnitas $x_1v_1 + ... + x_kv_k = v$.

Usando o exemplo acima como referência, como fazemos para verificar se $v = (1,1,0,0,0)$ pertence ao código C? Para pertencer, deve existir $x = (x_1,x_2,x_3,x_4)$ em $\mathbb{K}^k$ de modo que

$$T(x) = x_1(1,0,0,0,1) + x_2(0,1,0,0,0) + x_3(0,0,1,0,1) + x_4(0,0,0,1,1) = (1,1,0,0,0).$$

Ou seja

$x_1.1 + x_2.0 + x_3.0 + x_4.0 = 1$ se, e somente se $x_1 = 1$;
$x_1.0 + x_2.1 + x_3.0 + x_4.0 = 1$ se, e somente se $x_2 = 1$;
$x_1.0 + x_2.0 + x_3.1 + x_4.0 = 0$ se, e somente se $x_3 = 1$;
$x_1.0 + x_2.0 + x_3. + x_4.1 = 0$ se, e somente se $x_4 = 1$;
$x_1.1 + x_2.0 + x_3.1 + x_4.1 = 0$.

Note que os valores obtidos para $x_i, 1 \leq i \leq 4$, não satisfazem a última equação. Com isso concluímos que v não pertence a C.

Nesse exemplo específico, com $k = 4$ e $n = 5$ conseguimos determinar com certa facilidade que v não pertence a C. No entanto, como foi citado em capítulos anteriores, para que um código tenha maior eficiência, busca-se estruturar códigos com a quantidade de elementos M e distância mínima d "grandes" comparados com o comprimento n de seus elementos. A medida que n aumenta, o processo de verificação se torna maior. Nesses casos, determinar se o elemento pertence ou não ao código, demanda muito mais trabalho e capacidade computacional.

Agora vamos analisar a descrição do código C através do núcleo de uma transformação linear. Para isso, consideremos um subespaço C' tal que $C + C' = K^n$, em que a soma indica que C' é um complemento de C. Vamos considerar a aplicação linear

$$\begin{aligned} H : C \oplus C' &\longrightarrow \mathbb{K}^{n-k} \\ u \oplus v &\longmapsto v \end{aligned}$$

onde o núcleo é justamente C. Recorrer a este método torna um pouco mais simples verificar se v de $\mathbb{K}^n$ pertence ou não a C.

Exemplo 4.6. *Vamos considerar $\mathbb{Z}_3 = \{0,1,2\}$ e $C \subset \mathbb{Z}_3^4$ um código gerado por $v_1 = (1,0,1,1)$ e $v_2 = (0,1,1,2)$. Esse código pode ser representado como núcleo da transformação linear*

$$\begin{aligned} H : \mathbb{Z}_3^4 &\longrightarrow \mathbb{Z}_3^2 \\ (x_1,x_2,x_3,x_4) &\longmapsto (2x_1 + 2x_2 + x_3, 2x_1 + x_2 + x_4) \end{aligned}$$

Vamos verificar que $v_1 = (1,0,1,1)$ e $v_2 = (0,1,1,2)$ geram todo o núcleo de H, mostrando que esse conjunto é uma base do núcleo da transformação.

Aplicando H em v_1 e v_2, obtemos:

$$\begin{aligned} H(v_1) &= (2 \cdot 1 + 2 \cdot 0 + 1,\ 2 \cdot 1 + 0 + 1) = (2 + 0 + 1,\ 2 + 0 + 1) = (3,\ 3) = (0,\ 0); \\ H(v_2) &= (2 \cdot 0 + 2 \cdot 1 + 1,\ 2 \cdot 0 + 1 + 2) = (0 + 2 + 1,\ 0 + 1 + 2) = (3,\ 3) = (0,\ 0). \end{aligned}$$

Com isso, concluímos que v_1 e v_2 estão no núcleo.

Agora, dados $\alpha_1,\ \alpha_2 \in \mathbb{Z}_3$, verifiquemos que $\alpha_1(1,0,1,1) + \alpha_2(0,1,1,2) = 0$ se, e somente se $\alpha_1 = \alpha_2 = 0$. Podemos concluir isso apenas analisando a combinação linear das duas primeiras coordenadas de v_1 e de v_2: $\alpha_1 \cdot 1 + \alpha_2 \cdot 0 = 0$ implica que $\alpha_1 = 0$ e $\alpha_1 \cdot 0 + \alpha_2 \cdot 1 = 0$ implica que $\alpha_2 = 0$, logo esses vetores são linearmente independentes.

Do Teorema 2.60 temos que $dim\, \mathbb{Z}_3^4 = dim\, Nuc + dim\, \mathbb{Z}_3^2$. Então $4 = dim\, Nuc + 2$, o que implica que $dim\, Nuc = 2$.

Mostramos que v_1 e v_2 estão no núcleo da transformação e são linearmente independentes. Além disso, mostramos que o núcleo tem dimensão 2. Com isso, concluímos que $\{v_1, v_2\}$ é uma base do nucleo de H.

Portanto, o código C gerado por v_1 e v_2 pode ser descrito como o núcleo da transformação linear H.

Nesse exemplo específico, como temos uma base de C, podemos determinar todos seus elementos através de sua representação paramétrica $x_1v_1 + x_2v_2$, apenas analisando todas as combinações de x_1,x_2 variando em $\mathbb{Z}_3$. Mas o que importa para nós é que, dado qualquer elemento de $\mathbb{Z}_3^4$, fica fácil verificar se esse elemento pertence ao código C.

Exemplo 4.7. *Vamos verificar se* $(1,0,0,1,1,1)$ *e* $(0,1,0,1,0,1)$ *pertencem ao código* C*, onde* C *é núcleo da transformação linear*

$$\begin{aligned} T : \mathbb{Z}_2^6 &\longrightarrow \mathbb{Z}_2^3 \\ (x_1,...,x_6) &\longmapsto (x_1 + x_4, x_1 + x_2 + x_3 + x_5, x_1 + x_2 + x_6) \end{aligned}$$

$T(1,0,0,1,1,1) = (1+1, 1+0+0+1, 1+0+1) = (0,0,0)$*. Logo,* $(1,0,0,1,1,1)$ ***é um*** *elemento de* C*.*

$T(0,1,0,1,0,1) = (0+1, 0+1+0+0, 0+1+1) = (1,1,0)$*. Então,* $(1,0,0,1,1,1)$ ***não*** *é um elemento de* C*.*

Definição 4.8. *Seja* $\mathbb{K}^n$ *um espaço vetorial sob o corpo* $\mathbb{K}$ *e seja* $T : \mathbb{K}^n \to \mathbb{K}^n$ *uma isometria. Dizemos que* T *é uma* ***isometria linear*** *se* T *também for uma transformação liner.*

Definição 4.9. *Seja* $\mathbb{K}$ *um corpo finito. Dois códigos lineares* C *e* C' *são linearmente equivalentes se existir uma isometria linear* $T : \mathbb{K}^n \to \mathbb{K}^n$ *tal que* $T(C) = C'$*.*

Vamos considerar π uma permutação em $\{1,...,n\}$ e a aplicação T_π apresentada na Proposição 3.17. Além de T_π ser uma isometria, a aplicação também é uma isometria linear.

Agora, vamos considerar $f : \mathbb{K} \to \mathbb{K}$ uma bijeção.

Temos que a aplicação T_f^i apresentada na Proposição 3.16 será uma isometria linear se, e somente se, existe um elemento $c \in \mathbb{K}$ tal que $f(x) = cx$ para todo x. Ou seja, T_f^i é linear se f for linear. Com isso, conseguimos descrever uma forma de obter códigos linearmente equivalentes.

De forma geral, dois códigos lineares, contidos em um mesmo espaço vetorial, serão linearmente equivalentes se, e somente se, podemos obter um partindo do outro tomando como intermediação as duas seguintes operacões:

1. Multiplicar a i-ésima coordenada de todas as palavras do código por um escalar c pertencente a $\mathbb{K}$, com $c \neq 0$;

2. Aplicar uma permutação de $\{1,...,n\}$, reorganizando as coordenadas de todas as palavras do código.

Sejam $\mathbb{K}$ um corpo com q elementos e $C \subset \mathbb{K}^n$ um código linear. O parâmetro de um código é determinado pela terna (n,k,d), onde n é a dimensão do espaço onde C está contido, k é a dimensão de C e d é a distância mínima e o peso $\omega(C)$ de C.

Quando aplicamos uma isometria linear em C, obtendo o código C', linearmente equivalente, os parâmetros de C são preservados em C'. Eles têm a mesma estrutura essencial, ou seja, além dos pesos das palavras serem mantidos na imagem, a capacidade de detecção e de correção serão as mesmas.

4.2 Matriz geradora de um Código Linear

Dado um código linear C, vamos considerar $\beta = \{v_1,...,v_k\}$ sua base ordenada, onde $v_i = (v_{i_1}, v_{i_2},...,v_{i_n})$. Vamos dispor cada $v_i, 1 \leq i \leq k$ nas linhas de uma matriz G, da seguinte forma

$$G = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} = \begin{pmatrix} v_{1_1} & v_{1_2} & \cdots & v_{1_n} \\ v_{2_1} & v_{2_2} & \cdots & v_{2_n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{k_1} & v_{k_2} & \cdots & v_{k_n} \end{pmatrix}.$$

A matriz G é intitulada **matriz geradora** de C associada à base β.

Vamos considerar, agora, a seguinte transformação linear

$$\begin{aligned} T : \mathbb{K}^k &\longrightarrow \mathbb{K}^n \\ x &\longmapsto xG \end{aligned}$$

Escrevendo $x = (x_1,...,x_k)$, ao aplicarmos T nesse elemento de $\mathbb{K}^k$ teremos que

$$T(x) = xG = x_1v_1 + \cdots + x_kv_k.$$

Desse modo, podemos verificar que aplicando T em todo espaço $\mathbb{K}^k$, temos o próprio código C. Em outras palavras, $T(\mathbb{K}^k) = C$. Nesse caso, $\mathbb{K}^k$ é nosso código fonte, T é uma codificação e C é nosso código de canal (aquele que é enviado por algum meio de comunicação).

A construção da matriz G não depende exclusivamente de C, mas sim a partir da escolha de uma base β do código. Dito isso, vale lembrar que, dada uma base β de C, podemos definir outra base β' aplicando as seguintes operações em β:

- permutar dois elementos da base;
- multiplicar de um elemento da base por um escalar não nulo;
- trocar um elemento v da base por $v + \alpha u$, onde u também é um dos vetores que constituem a base β.

Trazendo um raciocínio análogo para a matriz G apresentada anteriormente, concluimos que, matrizes diferentes geradoras de um mesmo código podem ser obtidas uma da outra por uma sequência de operações do tipo:

Permutação de duas linhas;

Multiplicação de uma linha por um escalar diferente de zero;

Adição de outra linha que foi multiplicada por um múltiplo escalar diferente de zero.

Dito isso, se pensarmos o caminho oposto, determinamos uma forma de construir códigos lineares a partir de matrizes geradoras G. Para fazer isso basta apenas descrever uma matriz associada a uma base de um espaço vetorial, então, definimos C como imagem da transformação linear associada à matriz G.

Exemplo 4.10. *Seja* $\mathbb{K} = \mathbb{Z}_2$ *e seja*

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

Vamos considerar a transformação linear

$$\begin{array}{rccc} T: & \mathbb{Z}_2^3 & \longrightarrow & \mathbb{Z}_2^5 \\ & x & \longmapsto & xG \end{array}$$

Definimos a imagem de T sendo o código C. Tomemos o elemento (1,0,1) do código de fonte $\mathbb{Z}_2^3$.

$$T(1,0,1) = (1,0,1) \cdot \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (0,1,0,1,0).$$

A codificação da palavra $(1,0,1)$ do código de fonte fica $(0,1,0,1,0)$. Quando a ideia é codificar uma palavra do código de fonte, conseguimos fazer com certa facilidade. Agora, analisaremos o oposto: dado o elemento $(0,1,0,1,0)$ do código C (e da imagem da transformação T), qual a palavra do código fonte que foi transmitida inicialmente, ou, em outras palavras, qual o elemento $x = (x_1,\, x_2,\, x_3)$ tal que $T(x) = (0,1,0,1,0)$.

Para tal, temos que resolver o seguinte sistema de equação

$$(x_1,\, x_2,\, x_3) \cdot \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} = (0,1,0,1,0)$$

$x_1.1 + x_2.1 + x_3.1 = 0$
$x_2.1 + x_3.1 = 1$
$x_1.1 + x_3.1 = 0$

que tem como solução $(x_1\, x_2\, x_3) = (1,0,1)$.

Para esse sistema de equações, obtemos a solução de forma simples, mas de forma geral, se assumirmos uma matriz geradora G com entradas mais complexas, a resolução do mesmo pode se mostrar mais trabalhosa.

Levando isso em consideração, buscaremos uma maneira melhor para resolver esse tipo de problema. Note que, ao escalonarmos a matriz G, aplicando as operações elementares sobre as linhas, reescrevemos a mesma, obtendo a matriz

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

Agora, ao operarmos xG' temos

$$(x_1, x_2, x_3) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} = (x_1,\ x_2,\ x_3,\ x_2,\ x_3).$$

Isso significa que ao obtermos a palavra $(0,1,0,1,0)$ na imagem, torna-se mais fácil identificar que $(x_1,\ x_2, x_3) = (0,1,0)$ é a palavra do código fonte que foi enviada. Basta observarmos que $(0,1,0,1,0) = (x_1,\ x_2,\ x_3,\ x_2,\ x_3)$.

O sistema de equações proveniente da matriz G' é mais simples de se obter a solução. Podemos concluir que o trabalho com a matriz G' é simplificado se comparado com o trabalho que temos com a matriz G.

Definição 4.11. *Dizemos que uma matriz geradora G de um código C está na forma padrão se tivermos*

$$G = (Id_k|A),$$

onde Id_k é a matriz identidade de ordem $k \times k$ e A tem ordem $k \times (n-k)$.

No entanto, dado um código C gerado pela matriz G, nem sempre conseguimos deixar essa matriz na sua forma padrão. Vejamos a matriz que se segue:

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

Não conseguimos deixar G na forma padrão usando apenas as operações de escalonamento nas linhas. No entanto, se acrescentarmos ao conjunto de operações a permutação entre duas colunas, conseguimos determinar a matriz

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

Essa matriz, agora na forma padrão, não irá gerar o código C, mas sim o código C' equivalente a C.

Ou seja, considerando a matriz G, que não pode ser deixada na forma padrão, podemos afirmar que, ao acrescentarmos duas operações (a permutação entre duas colunas e a multiplicação de uma coluna por um escalar não nulo), juntamente com as operações de escalonamento das linhas da matriz, obtemos o resultado que se segue:

Teorema 4.12. *Dado um código C, existe um código equivalente C' com matriz geradora na forma padrão.*

Demonstração. Seja G uma matriz geradora de C. Queremos mostar que podemos colocar G na forma padrão aplicando as seguintes operações:

(L1) Permutação entre duas linhas;

(L2) Multiplicação de uma linha por um escalar não nulo;

(L3) Adição de um múltiplo escalar de uma linha a outra;

(C1) Permutação entre duas colunas;

(C2) Multiplicação de uma coluna por um escalar não nulo;

Seja

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}.$$

Como G é matriz geradora, temos que as linhas de G são linearmente independentes, ou seja, vai existir algum elemento na primeira linha ou na primeira coluna diferente de zero. Permutamos a linha ou a coluna de modo que este elemento fique na primeida linha e primeira coluna. Suponha sem perda de generalidade que $g_{11} \neq 0$. Agora, multiplicamos a primeira linha por g_{11}^{-1}, e denotando $g_{11}^{-1} \cdot g_{1j} = b_{1j}$, $2 \leq j \leq n$, obtemos a matriz

$$\begin{pmatrix} 1 & b_{12} & \cdots & b_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}.$$

Em seguida, iremos somar nas i-ésimas linhas, com $2 \leq i \leq k$, a primeira linha multiplicada por $-(g_{i1})$.

Aplicar essa operação nas i-ésimas linhas significa anular a primeira entrada de cada uma delas. Sendo assim, aplicamos a operação em todas essas linhas, exceto na primeira, obtendo assim a matriz

$$\begin{pmatrix} 1 & b_{12} & \cdots & b_{1n} \\ 0 & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & b_{k2} & \cdots & b_{kn} \end{pmatrix}.$$

Permutando as colunas, conseguimos reorganizar determinando um elemento b'_{22} diferente de zero na segunda linha e segunda coluna. Este elemento existe pois as linhas da matriz inicial são linearmente independentes. Agora, multiplicamos a segunda linha por $(b'_{22})^{-1}$, de modo que $b_{22}^{-1} \cdot b_{2j} = c_{2j},\ 1 \leq j \leq n$. Note que $c_{21} = 0$ não se altera. Vamos denotar por $c_{ij},\ 1 \leq i \leq k,\ 2 \leq j \leq n$ os outros elementos da matriz após essas operações. Obtemos então, a matriz

$$\begin{pmatrix} 1 & c_{12} & \cdots & c_{1n} \\ 0 & 1 & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & c_{k2} & \cdots & c_{kn} \end{pmatrix}.$$

Nesta última matriz, somamos nas i-ésimas linhas, $1 \leq i \leq k,\ i \neq 2$, a segunda linha multiplicada por $-(c_{2j}), 1 \leq j \leq n,\ j \neq 2$. Considere K_i o elemento da i-ésima linha na primeira coluna. Note que quando variamos i de 1 até k, fixando $j = 1$, temos $-(c_{21} \cdot b_{i1}) + K_i = K_i$. Ou seja, a primeira coluna não se altera com essa operação.

Como a ideia é anular todas as entradas da segunda coluna, exceto o elemento que está na segunda linha e segunda coluna, vamos analisar o que ocorre nesta coluna. Fixando $j = 2$ em $-(c_{2j} \cdot b_{ij}) + b_{ij} = c_{ij}$, isto é, $(c_{22} \cdot b_{i2}) + b_{i2} = c_{i2}$ temos $-(1 \cdot b_{i2}) + b_{i2} = c_{i2}$ pois $c_{22} = 1$. Isso implica em $c_{i2} = 0$.

Ao aplicarmos essa operação em qualquer que seja a linha (exceto na segunda linha), preserva-se a primeira entrada e anula-se a segunda desta linha. Sendo assim, determinamos a seguinte matriz

$$\begin{pmatrix} 1 & 0 & d_{13} & \cdots & d_{1n} \\ 0 & 1 & d_{23} & \cdots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & d_{k3} & \cdots & d_{kn} \end{pmatrix}.$$

.

Repetimos esse processo sucessivamente, até que G esteja na forma padrão.

□

Fechamos essa subseção acrescentando que, além de nos auxiliar com a codificação de um código fonte, o estudo de matrizes geradoras se faz necessário, principalmente, porque a analisarmos, conseguimos analisar a “aparência" do código que é gerado pela mesma. Basta

uma curta analise da matriz geradora que podemos determinar a dimensão do código, a dimensão do subespaço na qual o código está contido e também a quantidade de redundâncias que serão acrescentadas ao código fonte, através da codificação.

4.3 Códigos Duais

Antes de definirmos códigos duais, vamos relembrar alguns resultados já apresentados, que remetem a espaços vetoriais ortogonais.

Seja V um $\mathbb{K}$ espaço vetorial. Considerando $u = (u_1,...,u_n)$ e $v = (v_1,...,v_n)$ elementos de V, e o produto interno definido por $\langle u, v\rangle = u_1v_1 + ... + u_nv_n$, dizemos que u será ortogonal a v quando $\langle u, v\rangle = 0$. Agora, considerando o subespaço vetorial W de V, com produto interno bem definido, o complemento ortogonal de W, denotado por $W^\perp$, é o conjunto $W^\perp = \{v \in V \,|\, \langle w, v\rangle = 0, \forall\, w \in W\}$. Provamos também que, nessas condições, $W^\perp$ também será subespaço vetorial de V.

Trazendo esse conceito para nosso contexto, apresentamos a próxima definição.

Definição 4.13. *Seja $C \subset \mathbb{K}^n$ um código linear, o subespaço vetorial definido por*

$$C^\perp = \{v \in \mathbb{K}^n \,|\, \langle u, v\rangle = 0, \forall\, u \in C\}$$

é dito Código Dual do código C.

Lema 4.14. *Se $C \subset \mathbb{K}^n$ é um código linear, com matriz geradora G, então $x \in C^\perp$ se, e somente se, $Gx^T = 0$.*

Demonstração.

Seja G a matriz geradora de C dada por

$$G = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kn} \end{pmatrix}.$$

Então, temos que $v_i = (g_{i1}, g_{i2}, \cdots, g_{in}),\ 1 \leq i \leq k$, formam uma base para C.
Assim,

$$x \in C^\perp \iff \langle v_i, x\rangle = 0,\ 1 \leq i \leq k \iff Gx^T = \begin{pmatrix} \langle v_1, x\rangle \\ \langle v_2, x\rangle \\ \vdots \\ \langle v_k, x\rangle \end{pmatrix} = 0. \qquad \square$$

Proposição 4.15. *Seja $C \subset \mathbb{K}^n$ um código de dimensão k com matriz geradora na forma padrão $G = (Id_k \,|\, A)$. Então*

1. $dim\, C^{\perp} = n - k$;

2. $H = (-A^T \mid Id_{n-k})$ *é uma matriz geradora de* $C^{\perp}$.

Demonstração.

1. De acordo com o Lema 4.14, x pertence a $C^{\perp}$ se, e somente se, $Gx^T = 0$. Ou seja,

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & a_{1,k+1} & a_{1,k+2} & \cdots & a_{1,n} \\ 0 & 1 & \cdots & 0 & a_{2,k+1} & a_{2,k+2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & a_{k,k+1} & a_{k,k+2} & \cdots & a_{k,n} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Essa igualdade significa que

$$\begin{pmatrix} x_1 + a_{1,k+1}\, x_{k+1} + \cdots + a_{1,n}\, x_n \\ x_2 + a_{2,k+1}\, x_{k+1} + \cdots + a_{2,n}\, x_n \\ \vdots \\ x_k + a_{k,k+1}\, x_{k+1} + \cdots + a_{k,n}\, x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Separando em soma de matrizes de ordem $k \times 1$, temos

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} + \begin{pmatrix} a_{1,k+1}x_{k+1} + \cdots + a_{1,n}x_n \\ a_{2,k+1}x_{k+1} + \cdots + a_{2,n}x_n \\ \vdots \\ a_{k,k+1}x_{k+1} + \cdots + a_{k,n}x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

e, consequentemente,

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix} = - \begin{pmatrix} a_{1,k+1}x_{k+1} + \cdots + a_{1,n}x_n \\ a_{2,k+1}x_{k+1} + \cdots + a_{2,n}x_n \\ \vdots \\ a_{k,k+1}x_{k+1} + \cdots + a_{k,n}x_n \end{pmatrix}.$$

Assim

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix}_{n\times 1} = \begin{pmatrix} -(a_{1,k+1}x_{k+1} + a_{1,k+2}x_{k+2} + \cdots + a_{1,n}x_n) \\ -(a_{2,k+1}x_{k+1} + a_{2,k+2}x_{k+2} + \cdots + a_{2,n}x_n) \\ \vdots \\ -(a_{k,k+1}x_{k+1} + a_{k,k+2}x_{k+2} + \cdots + a_{k,n}x_n) \\ x_{k+1} \\ \vdots \\ x_n \end{pmatrix}_{n\times 1}$$

$$= x_{k+1}\begin{pmatrix} -a_{1,(k+1)} \\ -a_{2,(k+1)} \\ \vdots \\ -a_{k,(k+1)} \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{n\times 1} + x_{k+2}\begin{pmatrix} -a_{1,(k+2)} \\ -a_{2,(k+2)} \\ \vdots \\ -a_{k,(k+2)} \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}_{n\times 1} + \cdots + x_n\begin{pmatrix} -a_{1,n} \\ -a_{2,n} \\ \vdots \\ -a_{k,n} \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}_{n\times 1}$$

Logo

$$\begin{aligned} u_1 &= (-a_{1,k+1}, -a_{2,k+1}, \cdots, -a_{k,k+1}, 1, 0, \cdots, 0) \\ u_2 &= (-a_{1,k+2}, -a_{2,k+2}, \cdots, -a_{k,k+1}, 0, 1, \cdots, 0) \\ &\vdots \\ u_{n-k} &= (-a_{1,n}, \quad -a_{2,n}, \quad \cdots, -a_{k,n}, \quad 0, 0, \cdots, 1) \end{aligned}$$

gera $C^{\perp}$. É fácil ver que são linearmente independentes. Logo $dim\, C^{\perp} = n - k$.

2. Vimos na demonstração do item anterior que o conjunto $\{u_1, u_2, \cdots, u_{n-k}\}$ é uma base de $C^{\perp}$. Logo a matriz geradora de $C^{\perp}$ é

$$\begin{pmatrix} -a_{1,(k+1)} & -a_{2,(k+1)} & \cdots & -a_{k,(k+1)} & 1 & 0 & \cdots & 0 \\ -a_{1,(k+2)} & -a_{2,(k+2)} & \cdots & -a_{k,(k+2)} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{1,n} & -a_{2,n} & \cdots & -a_{k,n} & 0 & 0 & \cdots & 1 \end{pmatrix} = (-A^T | Id_{n-k}).$$

□

Lema 4.16. *Seja C um código de dimensão k em $\mathbb{K}^n$ e uma matriz geradora G. Uma matriz H de ordem $(n-k) \times n$, com coeficientes em $\mathbb{K}$ e com linhas linearmente independentes, é uma matriz geradora de $C^\perp$ se, e somente se, $G \cdot H^T = 0$.*

Demonstração.

$\Rightarrow$ Se H é uma matriz geradora de $C^\perp$, então as linhas de H são elementos de $C^\perp$ e também geram o espaço. Isso significa que para qualquer linha h_i,$1 \leq i \leq (n-k)$, da matriz H, tem-se $\langle c, h_i \rangle = 0$ para todo c pertencente ao código C. Em particular, vale para as linhas de G que são elementos de C. Assim,

$$(G \cdot H^T)_{i,j} = \langle g_i, h_j \rangle = 0,\ 1 \leq i \leq k,\ 1 \leq j \leq n-k.$$

Logo $G \cdot H^T = 0$.

$\Leftarrow G \cdot H^T = 0$ implica que o subespaço gerado pelas linhas de H está contido em $C^\perp$. Como este subespaço tem dimensão $n-k$ que é a mesma de $C^\perp$, segue que H é uma matriz geradora de $C^\perp$. □

Corolário 4.17. *Dado um código linear C, temos que $(C^\perp)^\perp = C$.*

Demonstração.

Considere G e H matrizes geradoras de C e $C^\perp$, respectivamente. Do resultado anterior temos que $G \cdot H^T = 0$. Aplicando a transposta em ambos lados da equação, temos que $(G\,H^T)^T = 0^T$ e, consequentemente, $H\,G^T = 0$. Essa igualdade está nos dizendo que G é matriz geradora do código $(C^\perp)^\perp$. Com isso, temos que G gera C e $(C^\perp)^\perp$, logo, $C = (C^\perp)^\perp$.

□

Proposição 4.18. *Seja C um código linear e H uma matriz geradora de $C^\perp$. Temos que c pertence ao código C se, e somente se, $Hc^T = 0$.*

Demonstração.

$\Rightarrow$ Vamos considerar c pertencente ao código C. Do corolário anterior temos que $C = (C^\perp)^\perp$, ou seja, c pertence a $(C^\perp)^\perp$. Logo, do Lema 4.14, $Hc^T = 0$.

$\Leftarrow Hc^T = 0$ significa que c pertence a $(C^\perp)^\perp$. Mas $(C^\perp)^\perp = C$, logo, c pertence ao código C. □

Dado um elemento c pertencente a $\mathbb{K}^n$, para verificarmos se c pertence ao código C faz-se necessário resolver o sistema de n equações com k incógnitas, assim como fizemos no Exemplo 4.10 da subseção anterior. Tal cálculo demanda capacidade computacional.

Perceba que agora, se temos a matriz H, conseguimos determinar de forma simplificada se c pertence ou não ao código C. Basta verificar se $Hc^T = 0$. No entanto, para isso, precisamos de um modo de determinar a matriz H partindo do que temos do código C e de sua matriz geradora G. Mas verificamos que o Item 2 da Proposição 4.15 nos apresenta justamente isso.

Definição 4.19. *Seja $C \subseteq \mathbb{K}_q^n$ um código linear de dimensão k. Uma matriz de verificação de paridade (ou **matriz teste de paridade**) para C é uma matriz H de dimensão $(n-k) \times n$, com entradas em $\mathbb{K}$ tal que :*

$$C = \{c \in \mathbb{K}_q^n \mid Hc^T = 0\}.$$

Definição 4.20. *Dados um código C com matriz teste de paridade H e um elemento qualquer $v \in \mathbb{K}_q^n$, chamamos Hv^T de **síndrome** de v.*

Basicamente, dado um elemento de $\mathbb{K}_q^n$, para sabermos se v pertence ou não ao código C, basta calcular sua síndrome. Se a sídrome for zero, ou seja, se $Hv^T = 0$, então v pertence ao código C.

Vamos refazer os cálculos do Exemplo 4.10 utilizando os últimos resultados e definições apresentados.

Seja C gerado pela matriz

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

A forma padrão do código C é dada pela matriz

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} = (Id_3|A), \text{ geradora do mesmo código.}$$

Queremos verificar se $v = (0,1,0,1,0) \in \mathbb{K}_2^5$ pertence ao código determinado por G. Para isso, precisamos calcular a matriz $H = (-A^T \mid Id_{n-k})$.

$$H = \begin{pmatrix} -(0) & -(1) & -(0) & 1 & 0 \\ -(0) & -(0) & -(1) & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Determinada a matriz H, basta calcularmos a síndrome de v:

$$Hv^T = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+1+0+1+0 \\ 0+0+0+0+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Com isso, concluímos que v pertence ao código C.

Note que, dado $x \in \mathbb{Z}_2$, tem-se $-(x) = x$. Ou seja, se estamos trabalhando com códigos sob o corpo $\mathbb{Z}_2$, calcular a matriz teste de paridade $(-A^T \mid Id_{n-k})$ é calcular $(A^T \mid Id_{n-k})$.

Por meio da matriz H também conseguimos extrair informações sobre o peso ω de um código C.

Proposição 4.21. *Seja H a matriz teste de paridade de um código C. Temos $\omega(C) \geq s$ se, e somente se, quaisquer $s-1$ colunas de H são linearmente independetes.*

Demonstração.

$\Rightarrow$ Vamos supor, por absurdo, que existe ao menos um conjunto com $s-1$ colunas de H que são linearmente dependentes.

Seja $\omega(C) = d$ determinado por x, ou seja, $\omega(x) = d$ e além disso, $x \neq 0$. Temos or hipótese que $\omega(x) \geq s$.

Como $x \in C$, temos que $Hx^T = 0$. Vamos escrever $x = (x_1,x_2,x_3,...,x_n)$ e

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} & ... & h_{1n} \\ h_{21} & h_{22} & h_{23} & ... & h_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{(n-k)\,1} & \cdots & \cdots & \cdots & h_{(n-k)\,n} \end{pmatrix}.$$

Ao operarmos Hx^T, obteremos a seguinte igualdade:

$$\begin{pmatrix} h_{11}x_1 + h_{12}x_2 + \cdots + h_{1n}x_n \\ h_{21}x_1 + h_{22}x_2 + \cdots + h_{2n}x_n \\ \vdots \\ h_{(n-k)1}x_1 + h_{(n-k)2}x_2 + \cdots + h_{(n-k)n}x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Escrevendo a matriz da esquerda como uma soma de matrizes, temos

$$\begin{pmatrix} h_{11}x_1 \\ h_{21}x_1 \\ \vdots \\ h_{(n-k)1}x_1 \end{pmatrix} + \begin{pmatrix} h_{12}x_2 \\ h_{22}x_2 \\ \vdots \\ h_{(n-k)2}x_2 \end{pmatrix} + \cdots + \begin{pmatrix} h_{1n}x_n \\ h_{2n}x_n \\ \vdots \\ h_{(n-k)n}x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

o que implica que

$$x_1 \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{(n-k)1} \end{pmatrix} + x_2 \begin{pmatrix} h_{12} \\ h_{22} \\ \vdots \\ h_{(n-k)2} \end{pmatrix} + \cdots + x_n \begin{pmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{(n-k)n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Como $\omega(x) = d$, essa combinação linear envolve exatamente d colunas de H, com coeficientes não nulos, ou seja, temos uma combinação linear não trivial de d colunas de H que resulta no vetor nulo. Isso mostra que essas d colunas são linearmente dependentes.

Mas da nossa suposição inicial, existe um conjunto de $s-1$ colunas que são linearmente dependentes. Uma contradição, pois $d = \omega(C) \geq s$.

$\Leftarrow$ Vamos supor que exista um vetor x não nulo pertencente a C de modo que $\omega(x) \leq s-1$. Ou seja, c possui no máximo $s-1$ coordenadas não nulas. Como x pertence a C, então $Hx^T = 0$. Ao analisarmos

$$x_1 \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{(n-k)1} \end{pmatrix} + x_2 \begin{pmatrix} h_{12} \\ h_{22} \\ \vdots \\ h_{(n-k)2} \end{pmatrix} + \cdots + x_n \begin{pmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{(n-k)n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

concluímos que H tem ao menos $s-1$ colunas que são linearmente dependentes, mas isso contradiz a hipótese que diz que quaisquer $s-1$ colunas são linearmente independentes. Portanto, não pode existir x com $\omega(x) \leq s-1$. Logo, $\omega(x) = \omega(C) \geq s-1$.

□

Teorema 4.22. *Seja H a matriz teste de paridade de um código C. Temos que o peso de C é igual a s se, e somente se, quaiquer $s-1$ colunas de H são linearmente independentes e existem s colunas de H linermente dependentes.*

Demonstração.

$\Rightarrow$ De fato, ao assumirmos $d = \omega(C) = s$, da Proposição 4.21, então quaiquer $s-1$ colunas de H são linearmente independentes. Por outro lado, existem s colunas de H que são linearmente dependentes, pois caso contrário, $\omega(C) \geq s+1$.

$(\Leftarrow)$ Suponha que todo conjunto com $s-1$ vetores coluna de H seja linearmente independente e que existam s colunas de H linearmente dependentes. Pela Proposição 4.21 temos $\omega(C) \geq s$. Não podemos ter $\omega(C) > s$, pois pela Proposição 4.21 teríamos que todo conjunto com s colunas de H é linearmente independente. O que é uma contradição com o fato de existir um conjunto com s colunas linearmente dependente. Logo $\omega(C) = s$.

□

Definição 4.23. *Seja $H \in \mathbb{K}^{(n-k)\times n}$ a matriz teste de paridade associada a um código linear $C \subseteq \mathbb{K}^n$. O posto da matriz H, denotado por $\mathrm{posto}(H)$, é definido como a dimensão do subespaço vetorial gerado pelas linhas de H. Ou seja, é o número máximo de colunas linearmente independentes da matriz H.*

Corolário 4.24. *(Cota de Singleton) Os parâmetros (n,k,d) de um código linear satisfazem a desigualdade $d \leq n-k+1$.*

Demonstração.

Se H é uma matriz teste de paridade, ela tem posto $n-k$. Mas também, do Teorema 4.22, $d=s$ implica que quaisquer $s-1$ colunas de H são linearmente independentes. Mas a quantidade de colunas linearmente independentes é igual ou menor ao posto, ou seja, $s-1 \leq n-k$, o que implica que $d=s \leq n-k+1$. Logo, segue a desigualdade $d \leq n-k+1$.

□

O estudo dos códigos duais nos fornece importantes ferramentas sobre a estrutura de um código linear, principalmente quando se trata da matriz teste de paridade. Na próxima subseção trataremos do processo de decodificação de códigos lineares. Veremos também o papel central, tanto da matriz teste de paridade quanto da síndrome, para o processo de detecção e correção de erros.

4.3.1 Decodificação

Denominamos como decodificação o processo de detecção e correção de erros em uma determinada palavra recebida, com objetivo de recuperar a mensagem original transmitida.

Para começarmos, vamos definir vetor erro. Considere que ao transmitir uma mensagem c, os ruídos distorcem a mesma, chegando ao destino a mensagem r. Chamaremos de erro e justamente a diferença de c para r, ou seja,

$$e = r - c.$$

Digamos que $(1,1,0,1,0,1)$ sobre $\mathbb{Z}_2$ foi transmitido e, ao passar pelo canal de transmissão, a mensagem recebida é $(1,1,0,1,0,0)$. Então o vetor erro será justamente

$$e = (1,1,0,1,0,0) - (1,1,0,1,0,1) = (0,0,0,0,0,1).$$

Note que o peso de e é justamente a quantidade de entradas que foram alteradas por ruídos. Além disso, a coordenada não nula mostra justamente em qual entrada a palavra foi alterada.

Agora, de forma geral, vamos considerar H a matriz teste de paridade de C. Temos que

$$He^T = H(r-c)^T = Hr^T - Hc^T.$$

Como Hc^T é nulo, resta que $He^T = Hr^T$, ou seja, a síndrome de e é igual a síndrome de r.

Por fim, tomemos $e = (\alpha_1,...,\alpha_n)$, $\alpha_i \in \mathbb{K}$, e $h_i = (h_{ij})$ a i-ésima coluna de H, $1 \leq j \leq n-k$. Então

$$He^T = \alpha_1 h_1 + \alpha_2 h_2 + \cdots + \alpha_n h_n$$
$$= \alpha_1 \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{(n-k)1} \end{pmatrix} + \alpha_2 \begin{pmatrix} h_{12} \\ h_{22} \\ \vdots \\ h_{(n-k)2} \end{pmatrix} + \cdots + \alpha_n \begin{pmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{(n-k)n} \end{pmatrix} = Hr^T.$$

Note, na equação acima, que se $e = (0,...0)$, a síndrome é o vetor nulo, ou seja, não foram cometidas alterações na transmissão. Vamos supor agora que $e = (\alpha_1,0,...,0)$, $\alpha_1 \neq 0$. Todas colunas de H, exceto a primeira, serão anuladas, além disso, a síndrome de He^T terá como resultado um múltiplo da primeira coluna - em nosso caso, como estamos trabalhando sobre $\mathbb{Z}_2$, a síndrome será uma vez a primeira coluna, ou seja, a própria coluna em si. Em outras palavras, para verificar se houve alterações em uma palavra, acometido pelo canal de transmissão, basta calcular a síndrome da palavra recebida: se a síndrome for o vetor nulo, então a mensagem chegou corretamente; se a síndrome for diferente do vetor nulo, verificamos através da combinação linear das colunas de H, quais estão envolvidas na combinação linear que gerou a síndrome. Isso nos permite identificar qual coordenada da palavra recebida que foi alterada.

Para finalizar, é importante perceber que a conclusão obtida se deu sem necessariamente conhecermos c, ou seja, apenas tendo e e r, que é justamente o que precisamos.

Lema 4.25. *Seja C um código linear em $\mathbb{K}^n$ com capacidade de correção κ. Se $r \in \mathbb{K}^n$ e $c \in C$ são tais que $d(c,r) \leq \kappa$, então existe um único vetor e com $\omega(e) \leq \kappa$, cuja síndrome é igual à síndrome de r e tal que $c = r - e$.*

Demonstração. Como $e = r - c$, pelo Item 1 da Proposição 4.4, $\omega(e) = w(r - c) = d(r,c)$. Da hipótese, $d(c,r) \leq \kappa$, logo, $\omega(e) \leq \kappa$.

Nos basta provar a unicidade. Vamos considerar que existem $e = (\alpha_1,...,\alpha_n)$ e $e' = (\alpha'_1,...,\alpha'_n)$ tais que $\omega(e) \leq \kappa$ e $\omega(e') \leq \kappa$, ambos com a mesma síndrome de r. Então, sendo H a matriz teste de paridade, $He^T = H(e')^T$ implica que

$$H(e-e')^T = (\alpha_1 - \alpha'_1) \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{(n-k)1} \end{pmatrix} + (\alpha_2 - \alpha'_2) \begin{pmatrix} h_{12} \\ h_{22} \\ \vdots \\ h_{(n-k)2} \end{pmatrix} + \cdots + (\alpha_n - \alpha'_n) \begin{pmatrix} h_{1n} \\ h_{2n} \\ \vdots \\ h_{(n-k)n} \end{pmatrix} = 0,$$

onde no máximo $\kappa + \kappa = 2\kappa$ colunas não são anuladas, o que nos dá uma relação de dependência linear entre $2\kappa \leq d - 1$ colunas de H. Como do Teorema 4.22 quaisquer $d - 1$ colunas

de H são linearmente independetes, temos que as 2κ colunas da combinação linear acima são linearmente independetes, ou seja, $\alpha_i = \alpha_i'$ para todo i, logo, $e = e'$.

□

A partir de tudo que foi apresentado, mostraremos na prática o procedimento para identificar e corrigir erros, com base na estrutura algébrica dos códigos lineares.

Exemplo 4.26. *Vamos considerar o código do robô C com matriz geradora*

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

e matriz teste de paridade

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

Suponha que numa transmissão seja recebida a palavra $r = (1,0,1,0,0)$, então

$$Hr^T = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Note que a síndrome é exatamente $1 \cdot h_4$, onde h_i é a i-ésima coluna da matriz teste de paridade. Isso significa que um erro foi introduzido exatamente na quarta coordenada do vetor enviado. Da relação $e = r - c$ concluímos que $c = (1,0,1,0,0) - (0,0,0,1,0) = (1,0,1,1,0)$.

Note que o exemplo acima apresenta o processo de decodificação para o caso em que $\omega(e) = 1$, ou seja, é o caso específico em que exatamente uma coordenada é modificada pelo canal de transmissão.

Com isso, a seguir estabelecemos o algoritmo de decodificação em códigos corretores de um erro.

Seja H a matriz teste de paridade do código C e seja r um vetor recebido. Vamos cosnsiderar $d \geq 3$.

1. Calcule Hr^T;
2. Se $Hr^T = 0$, aceitamos r como a palavra transmitida;
3. Se $Hr^T \neq 0$, compare Hr^T com as colunas de H;

4. Se existirem i e α tais que $Hr^T = \alpha h_i$, para $\alpha \in \mathbb{K}$, então e é a n-upla com α na posição i e zeros nas outras posições. Corrija r tomando $c = r - e$;

5. Se no passo anterior não for possível determinar $Hr^T = \alpha h_i$ comparando a síndrome com as colunas de H, significa que mais de um erro ocorreu;

O exemplo que se segue, é o caso em que não se encontra $Hr^T = s^T = h_i$.

Exemplo 4.27. *Seja o código de dimensão 4 pertencente a $\mathbb{Z}_2^7$ gerado pela matriz*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \text{ com matriz teste de paridadade } H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Suponhamos que ao transmitir $c = (0,1,0,1,1,1,1)$, a palavra recebida é $r = (0,1,0,1,1,0,0)$. Calculando Hr^T teremos

$$\begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0+0+0+1+1+0+0 \\ 0+1+0+0+0+0+0 \\ 0+0+0+1+0+0+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}.$$

A síndrome Hr^T é diferente de qualquer coluna da matriz H. Note que

$$Hr^T = 1h_6 + 1h_7.$$

Logo, poderíamos concluir que as alterações ocorreram na sexta e sétima coordenada, ou seja, $e = (0,0,0,0,0,1,1)$. Verificando $e = r - c$ concluímos que

$$c = (0,1,0,1,1,0,0) - (0,0,0,0,0,1,1) = (0,1,0,1,1,1,1).$$

Nesse exemplo apresentado, sabemos que as alterações ocorreram na sexta e na sétima coordenada porque assumimos isso desde o princípio, ou seja, sabíamos que $Hr^T = h_6 + h_7$. Mas no contexto de decodificação, não sabemos previamente onde o erro

ocorreu. Em vez disso, calculamos a síndrome e então tentamos encontrar uma combinação de colunas (ou uma única coluna) de H que corresponda a ela. A análise inicial demonstrou que $Hr^T = h_6 + h_7$. A questão é se essa é a única ou a mais provável combinação, especialmente em códigos que corrigem múltiplos erros. No entanto, observe que $Hr^T = h_3 + h_4$. Este é um problema pois o decodificador não conseguiria distinguir qual par de erros de fato ocorreu. Isso significa que o código não é capaz de corrigir dois erros nesse cenário, pois há ambiguidade. Isso mostra que o código apresentado não é eficiente.

Isso ocorre por alguns motivos. O principal se da pelos parâmetros do código. Outro motivo é o modo como as redundâncias foram introduzidas. Usaremos o Corolário 4.24 - Cota de Singleton - para que fique esclarecido.

De acordo com a Cota de Singleton, $d \leq n - k + 1$. Vamos analisar essa inequação com os parâmetros do nosso código.

$$d \leq n - k + 1 = 7 - 4 + 1 = 4.$$

Isso significa que qualquer código de dimensão $k = 4$, contido em um espaço de dimensão $n = 7$, pode ter, no máximo, distância mínima igual a 4 e capacidade de correção no máximo

$$\kappa \leq \left[\frac{4-1}{2}\right] = 1.$$

Em nosso caso, o código gerado pela matriz G tem distância mínima igual a 2, ou seja,

$$\kappa = \left[\frac{2-1}{2}\right] = 0\,.$$

Com isso, concluímos que o código do exemplo acima é ineficiente por sua própria estrutura. Além disso, existem outras matrizes geradoras que acrescentam as redundâcias de forma mais eficiente, resultando na geração de códigos mais eficazes.

Verificaremos agora como ocorre de forma geral para o caso que ocorrem exatamente duas alterações na mensagem enviada.

Exemplo 4.28. *Seja o código C de dimensão k contido em $\mathbb{Z}_2^n$, com distância mínima $d \geq 5$ e com capacidade de correção de $\dfrac{d-1}{2} \geq 2$ erros. Escrevemos*

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & \cdots & h_{1n} \\ h_{21} & h_{22} & h_{23} & \cdots & h_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{(n-k)1} & h_{(n-k)2} & h_{(n-k)3} & \cdots & h_{(n-k)n} \end{bmatrix}$$

e denotamos por h_i a i-ésima coluna da matriz teste de paridade H.

Montaremos uma tabela com todas as $\binom{n}{2}$ combinações de somas de $\dfrac{5-1}{2} = 2$ colunas de H, pois estamos assumindo que essa será a quantidade de alterações cometidas.

Tabela 3 – Todas as somas de duas colunas da matriz H

Par de colunas
$h_1 + h_2$
$h_1 + h_3$
$\vdots$
$h_1 + h_n$
$h_2 + h_3$
$h_2 + h_4$
$\vdots$
$h_{n-1} + h_n$

Fonte: Autor.

Vamos supor agora que ao transmitirmos a palavra c, recebemos a palavra r. Ao calcularmos $Hr^T = He^T$, não poderemos comparar a síndrome com as colunas de H, pois estamos considerando que foram cometidos dois erros. No entanto, podemos comparar com a Tabela 3 que montamos. Vamos supor que $Hr^T = h_i + h_j$, logo, as modificações ocorreram na i-ésima e na j-ésima entrada. Assim, com o vetor de erro e determinado, a mensagem original pode ser recuperada através da relação $e = r - c$.

Para análise do caso em que são cometidos até três erros, é necessário definir um código com distância mínima $d \geq 7$, pois assim a capacidade de correção do mesmo será dado por $\kappa \geq \dfrac{7-1}{2} = 3$. Depois, dada a matriz teste de paridade H, montar uma tabela com o cálculo de todas as $\binom{n}{3} + \binom{n}{2} + \binom{n}{1}$ combinações de soma das suas colunas. Ao receber r com até três modificações, calculamos sua síndrome e comparamos com os resultados dispostos na tabela.

De forma geral, dado um código com distância mínima d, com capacidade de correção $\kappa = \dfrac{d-1}{2}$ e com matriz teste de paridade H, montamos uma tabela com $\displaystyle\sum_{i=1}^{\kappa} \binom{n}{i}$ somas das colunas de H. Ao calcularmos a síndrome da mensagem recebida, comparamos a mesma com as somas dispostas na tabela. Assim, conseguimos identificar quais entradas da mensagem original foram modificadas.

No somatório acima podemos ver mais uma vez a importância da construção de códigos com n e d relativamente grandes, pois os mesmos também terão uma capacidade de correção relativamente grande.

Por fim, uma consideração a ser feita é que o procedimento apresentado acima se torna exaustivo em códigos mais sofisticados. Em outras palavras, seria necessário investimento computacional para o uso do mesmo.

Existem outros mecanismos mais práticos que demandariam menos gastos. Esses procedimentos são derivados do método apresentado. No entanto, seria necessário um estudo aprofundado de outros conceitos da álgebra, como por exemplo as classes laterais de um grupo. Com isso, encerramos nosso estudo de códigos lineares, expondo os principais aspectos do tema e estabelecendo uma base sólida para investigações e estudos futuros.

5 CONCLUSÃO

Mesmo que o estudo de Códigos Corretores de Erros não faça parte da matriz do curso de Licenciatura em Matemática, seu estudo enfatiza o conhecimento matemático, tendo em vista que o mesmo exigiu a aplicação de diversos conceitos fundamentais vistos na graduação, como aritmética modular, estruturas algébricas, corpos finitos, espaços vetoriais, transformações lineares, etc. Este aprofundamento mostra um pouco das várias linhas que a Matemática oferece. Ao nos referirmos a Claude Shannon e Richard Hamming, e levando em consideração seus estudos e pesquisas, concluímos que eles deram origem a um novo objeto de estudo - os códigos corretores de erros - e revolucionaram toda a área da Teoria da Informação. Para finalizar, destacamos a importância da Matemática na solução de problemas e no desenvolvimento das diversas áreas de estudo, como a tecnologia, demonstrando seu papel fundamental na evolução científica e tecnológica.

REFERÊNCIAS

BAHIA, F. Um primeiro curso sobre códigos corretores de erros. *In*: ENCONTRO REGIONAL DE MATEMÁTICA APLICADA E COMPUTACIONAL. 2010. **Anais [...]** [*S.l.*]: ERMAC, 2010. p. 149–169.

COELHO, F. U.; LOURENÇO, M. L. **Um Curso de Álgebra Linear**. São Paulo: edusp, 2013.

HEFEZ, A.; VILLELA, M. L. T. **Códigos Corretores de Erros**. Rio de Janeiro: IMPA, 2008.

MILLIES, F. C. P. Breve introdução à teoria dos códigos corretores de erros. *In*: COLÓQUIO DE MATEMÁTICA DA REGIÃO CENTRO-OESTE. 2009, Campo Grande. **Anais [...]** Campo Grande: Sociedade Brasileira de Matemática (SBM), 2009.

SABADIN, G. A. P. **Códigos corretores de erros**. 2019. 150 p. Dissertação (Mestrado) — Programa de Pós-Graduação em Matemática, Universidade Federal de Santa Catarina Florianópolis 2019.

VENTURA, J. **Notas de Combinatória e Teoria de Códigos**. [*S.l.*]: , 2013. Disponível em: https://www.math.tecnico.ulisboa.pt/~jventura/CTC/CTCnotas.pdf. Acesso em: 02 jun. 2025.