UNIVERSIDADE FEDERAL DE VIÇOSA

AMANDA PONTES DE OLIVEIRA ORNELAS

UMA ABORDAGEM COMPUTACIONAL AOS CÓDIGOS DE GRUPO COM O AUXÍLIO DO GAP

VIÇOSA - MINAS GERAIS
2021

AMANDA PONTES DE OLIVEIRA ORNELAS

UMA ABORDAGEM COMPUTACIONAL AOS CÓDIGOS DE GRUPO COM O AUXÍLIO DO GAP

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Matemática, para obtenção do título de *Magister Scientiae.*

Orientadora: Marinês Guerreiro

VIÇOSA - MINAS GERAIS
2021

Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa

T
O74a
2021

Ornelas, Amanda Pontes de Oliveira, 1996-
Uma abordagem computacional aos códigos de grupo com o auxílio do GAP / Amanda Pontes de Oliveira Ornelas. – Viçosa, MG, 2021.
95 f. : il. ; 29 cm.

Orientador: Marinês Guerreiro.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 94-95.

1. Códigos corretores de erros (Teoria da informação). 2. Álgebras de grupo. I. Universidade Federal de Viçosa. Departamento de Matemática. Programa de Pós-Graduação em Matemática. II. Título.

CDD 22. ed. 519.7

Bibliotecário(a) responsável: Renata de Fatima Alves CRB6/2578

AMANDA PONTES DE OLIVEIRA ORNELAS

UMA ABORDAGEM COMPUTACIONAL AOS CÓDIGOS DE GRUPO COM O AUXÍLIO DO GAP

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Matemática, para obtenção do título de *Magister Scientiae.*

APROVADA: 26 de julho de 2021.

Assentimento:

Amanda Pontes de O. Ornelas.

Amanda Pontes de Oliveira Ornelas
Autora

Marinês Guerreiro

Marinês Guerreiro
Orientadora

Agradecimentos

Agradeço primeiramente a Deus que esteve a minha frente em todos os momentos desta trajetória me fortalecendo todas as vezes em que eu fraquejava. Através da Capela da UFV e com a ajuda de todas as pessoas que ali conheci e que foram sinais de Deus na minha vida, pude renovar as minhas forças e seguir firme nesta caminhada.

Agradeço a minha orientadora, Marinês Guerreiro, por toda paciência, força, compreensão e ensinamentos neste tempo. Por todas as vezes em que me fez ver a vida e o mundo com outros olhos. Marinês se tornou muito além de professora e orientadora, uma grande amiga que tenho certeza que levarei para toda a vida.

Agradeço aos professores que estiveram comigo ao longo do curso e em tantos momentos me fortaleceram, em especial a Sônia Fernandes, que me ajudou a encontrar o caminho pelo qual seguir.

Agradeço aos meus familiares que estiveram comigo durante este caminho, em especial aos meus pais Ademilson Ornelas e Sonia Ornelas, meu irmão Handerson Ornelas e a Wallace Barboza.

Agradeço aos amigos que estiveram comigo ao longo deste curso e que juntos crescemos academicamente e humanamente, em especial a Douglas José de Souza que sempre se disponibilizou a me ajudar nesta trajetória, a Cíntia Coelho dos Santos e Angie Yurani Puentes Soler, que foram grandes apoios para que eu chegasse até aqui.

Agradeço a CAPES e a FAPEMIG pelo apoio financeiro desta pesquisa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"Eu vi o meu limite vir diante de mim. Eu enfrentei batalhas que eu não venci. Mas o troféu não é de quem não fracassou. Eu tive muitas quedas, mas não fiquei no chão. (...) E hoje eu sou quem eu sou pois Sua mão me acompanhava. Mas eu sei, não é o fim, é só o começo da jornada. Eu abro o meu coração pra minha nova história".

(Só o começo - Vocal Livre)

Resumo

ORNELAS, Amanda Pontes de Oliveira, M.Sc., Universidade Federal de Viçosa, julho de 2021. **Uma abordagem computacional aos códigos de grupo como auxílio do GAP.** Orientadora: Marinês Guerreiro.

Esse trabalho utiliza álgebras de grupo para o estudo de Códigos Corretores de Erros. Como, no entanto, a partir de um certo momento torna-se quase impossível a realização das contas à mão, optamos por utilizar uma ferramenta computacional como auxílio, o GAP (Groups, Algorithms, Programming), um sistema de álgebra computacional gratuito. No decorrer do trabalho, analisamos, principalmente, os códigos de grupo nas álgebras de grupo dos grupos simétricos S_3 e S_4 sobre o corpo finito F_5. Assim, percebemos que, nestes casos, os códigos não abelianos possuem parâmetros melhores que os códigos abelianos, mostrando e enfatizando a importância do estudo de códigos não abelianos. Além disso, estudamos um processo de decodificação e percebemos que há uma constante busca por algoritmos mais eficientes.

Palavras-chave: Códigos Corretores de Erros. Álgebras de Grupo.

Abstract

ORNELAS, Amanda Pontes de Oliveira, M.Sc., Universidade Federal de Viçosa, July, 2021. **A computational approach to group codes as an aid to GAP.** Orientadora: Marinês Guerreiro.

In this work we use group algebras to study Error Correcting Codes. However, as from a certain point onwards, it becomes almost impossible to carry out the calculations by hand, we chose to use a computational tool to help us, GAP (Groups, Algorithms, Programming), which is a free computer algebra system. In the course of the work, we analyzed mainly the group codes in the group algebras of the symmetric groups S_3 and S_4 over the finite field F_5. Thus, we realize that, in these cases, the non-abelian codes have better parameters than the abelian codes, showing and emphasizing the importance of studying non-abelian codes. In addition, we studied a decoding process and noticed that there is a constant search for more efficient algorithms.

Keywords: Error Correction Code. Group Algebras.

Sumário

Introdução

No processo de transmissão de dados podem ocorrer problemas no caminho, chamados ruídos, que fazem com que a mensagem recebida pelo receptor seja diferente da mensagem enviada. Assim, o objetivo da Teoria de Códigos Corretores de Erros é aplicar métodos para detectar e corrigir esses ruídos, permitindo que a mensagem chegue corretamente ao destinatário.

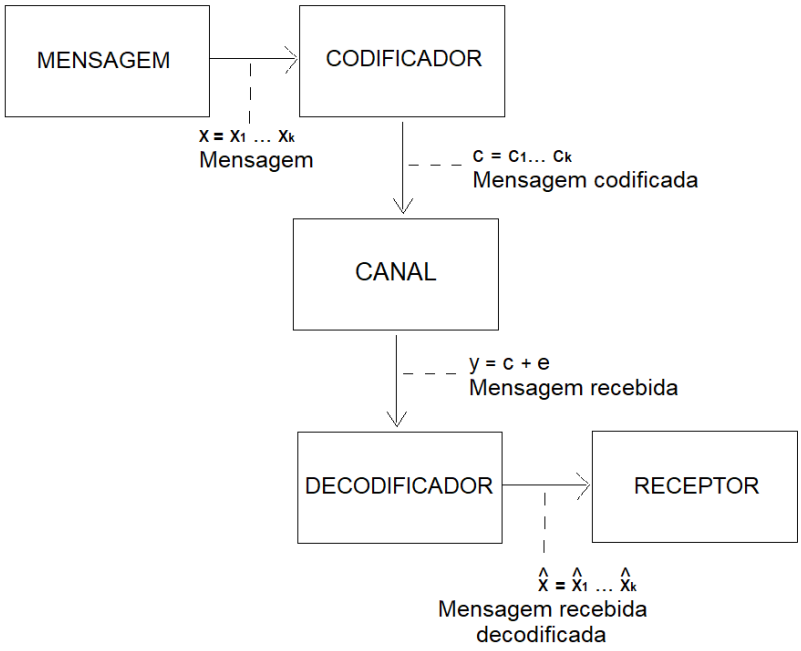
Essa teoria surgiu na década de quarenta do século XX, no Laboratório Bell de Tecnologia. No ano de 1947, Richard W. Hamming trabalhava nesse laboratório com máquinas que executavam tarefas numéricas complexas usando programas gravados em cartões perfurados. A leitura desses cartões pelo computador possibilitava detectar erros de digitação. Foi a partir disso que Hamming pensou que, se o computador era capaz de detectar um erro, talvez ele também fosse capaz de localizar sua posição e corrigí-lo. Assim, Hamming desenvolveu um código que permitia a detecção de até dois erros e a correção de um erro, se esse for o único. Durante os anos seguintes, ele publicou vários artigos internos do Laboratório Bell em que refletia sobre a possibilidade de criação de códigos melhores. Respondendo a essas reflexões, C. E. Shannon publicou um artigo que deu início a dois novos campos de pesquisa em matemática: Teoria de Códigos e Teoria da Informação [12].

Segundo à Teoria de Códigos, a transmissão de uma mensagem ocorre da seguinte maneira: uma mensagem x será enviada para um receptor, para isso, ela passa por um decodificador e é transformada numa mensagem codificada c. Assim, c é enviado ao receptor através do canal. Antes de chegar ao receptor, a mensagem passa por um decodificador, que tem como objetivo decodificar a mensagem detectando e corri-

gindo os possíveis ruídos/erros ocoridos no caminho. Por fim a mensagem decodificada é entregue ao receptor.

A Figura 1 é um esquema do processo explicado de transmissão de uma mensagem.

Figura 1: Processo de Transmissão de uma mensagem



Esse trabalho tem como referência principal a tese [17]. No início de cada capítulo apresentaremos as principais referências utilizadas no mesmo.

O contexto desse trabalho é utilizar álgebras de grupo para o estudo de códigos corretores de erros. Um ideal de uma álgebra de grupo de um grupo G é dito um *código de grupo* (ou um G-*código*). Para um grupo abeliano A, um A-código é dito um *código abeliano*. No entanto, pode acontecer de um G-código, para um grupo G não abeliano, ser equivalente a um A-código, para A um grupo abeliano.

Nessa dissertação buscamos fazer uma comparação entre códigos abelianos e códigos não abelianos relacionados a álgebras de grupo semissimples, para assim verificar

se vale a pena utilizar códigos não abelianos, já que esses são mais difíceis de lidar que os abelianos. Uma classe de grupos não abelianos para os quais os códigos de grupo são equivalentes a códigos abelianos é aquela na qual os grupos podem ser decompostos como produto de dois subgrupos abelianos. Estes grupos são chamados *grupos decomponíveis*. Neste trabalho identificamos um grupo de ordem 24 não decomponível, em cuja álgebra de grupo existem códigos não abelianos com bons parâmetros. Além disso, nesta pesquisa estudamos um processo de decodificação e a busca por algoritmos que decodifiquem de maneira eficiente.

Um outro objetivo deste trabalho foi utilizar uma ferramenta computacional para a realização dos cálculos necessários, que a partir de um dado momento são quase impossíveis de serem realizados à mão. O software escolhido para esta finalidade foi o GAP.

O GAP é um sistema computacional para álgebra discreta, com enfase, principalmente, no contexto da Teoria de Grupos. Seu nome é uma abreviatura de *Groups, Algorithms, Programming*. Esse sistema possui uma linguagem de programação, diversas funções e bibliotecas. O GAP é gratuito e apesar de não possuir interface gráfica, oferece manuais bem detalhados que auxiliam para a compreensão do seu funcionamento. [1]

Neste trabalho, para a realização dos cálculos, utilizamos o pacote do GAP chamado "Wedderga - Wedderburn Decomposition of Group Algebras" apresentado em [3].

No Capítulo 1 apresentamos resultados e definições preliminares da Teoria de Códigos Corretores de Erros e de Anéis de Grupo.

No Capítulo 2 apresentamos alguns resultados de classificação de grupos decomponíveis. Apresentamos também as relações entre os G-códigos abelianos sobre um corpo E e sobre um subcorpo F de E. Além disso, neste capítulo temos a teoria utilizada pelo GAP para o cálculo dos idempotentes primitivos centrais.

No Capítulo 3 apresentamos os códigos em F_5S_4 analisando quais são abelianos e quais são não abelianos e fazendo um comparativo dos seus parâmetros.

Por último, no Capítulo 4, apresentamos um processo de decodificação e damos

um exemplo de codificação e decodificação de uma palavra de um código em F_5S_3. Finalizamos comparando e analisando os parâmetros dos códigos dessa álgebra de grupo.

Capítulo 1

Preliminares

Nesse capítulo abordamos alguns temas que serão importantes para a compreensão do trabalho. Entre eles, encontram-se os anéis e álgebras de grupo e alguns tipos de códigos (lineares, duais, cíclicos e códigos de grupo). Além disso, alguns desses temas já serão apresentados com a utilização do GAP para a familiarização do leitor com o mesmo.

1.1 Anéis de Grupo

Essa seção aborda alguns dos principais conceitos e teoremas relacionados aos anéis de grupo. Os tópicos aqui apresentados, bem como a notação utilizada, se referem aos Capítulos 2 e 3 do livro [14] e ao livro [13]. O tema aqui apresentado será importante para a compreeensão dos capítulos seguintes, nos quais utilizamos as álgebras de grupo para construir códigos.

Definição 1.1.1. *Seja G um grupo (não necessariamente finito) e R um anel com unidade. O conjunto*

$$RG = \left\{ \alpha = \sum_{g \in G} a_g g \ / \ g \in G, a_g \in R \text{ e } a_g = 0 \text{ quase todo } g \in G \right\}$$

é chamado ***anel de grupo****. Os elementos de RG são combinações lineares formais com um número*

finito de coeficientes não nulos. Se R for um corpo, RG é chamado ***álgebra de grupo****.*

Perceba que o anel de grupo pode ser visto como um módulo com base G. Analogamente, a álgebra de grupo RG, vista como espaço vetorial, tem G como base. Assim,

$$dim\, RG = |G|.$$

Para definir as operações no anel de grupo, considere $\alpha = \sum_{g\in G} a_g g$ e $\beta = \sum_{g\in G} b_g g$ elementos de RG e $\lambda \in R$. Assim,

$$\alpha + \beta = \sum_{g\in G} a_g g + \sum_{g\in G} b_g g = \sum_{g\in G} (a_g + b_g)g$$

$$\alpha\beta = (\sum_{g\in G} a_g g)(\sum_{g\in G} b_g g) = \sum_{g,h\in G} a_g b_h gh.$$

A ação de R em RG ocorre da seguinte maneira: para $\lambda \in R$, temos

$$\lambda(\sum_{g\in G} a_g g) = \sum_{g\in G} (\lambda a_g)g.$$

Além disso, se RG é um anel com unidade, então $1 = \sum_{g\in G} u_g g$, com $u_g = 0$, para todo $g \neq e_G$ e $u_{e_G} = 1_R$.

Definição 1.1.2. *Seja RG um anel de grupo do grupo G sobre um anel R.*

Definimos o ***suporte*** *de $\alpha \in RG$ como o subconjunto de elementos em G que efetivamente aparecem na expressão de α, isto é,*

$$supp(\alpha) \;=\; \{g \in G \;:\; a_g \neq 0\}.$$

Observemos os seguintes exemplos feitos no GAP para entender melhor os conceitos apresentados até o momento. Para aplicar o conteúdo dessa sessão no GAP podemos usar dois pacotes: Laguna e Wedderga. O primeiro é utilizado para álgebras p-modulares,

isto é, quando F é um corpo de característica p e G é um p-grupo finito, e o segundo é utilizado para álgebras semissimples. As definições apresentadas até o momento são referentes a anéis e álgebras de grupo gerais, por isso os resultados independem do pacote utilizado.

Código 1.1: Exemplo e contraexemplo de Álgebra de Grupo no GAP

```
gap> LoadPackage("laguna");
gap> K := GF( 2 );
GF(2)
gap> Elements(K);
[ 0*Z(2), Z(2)^0 ]
gap> G := DihedralGroup( 16 );
<pc group of size 16 with 4 generators>
gap> KG := GroupRing( K, G );
<algebra-with-one over GF(2), with 4 generators>
gap> IsGroupAlgebra( KG );
true
gap> IsGroupAlgebra( GroupRing( Integers,DihedralGroup( 16 )));
false
```

No exemplo acima, construimos o anel de grupo do grupo $\mathbb{F}_2 D_{16}$ e pedimos para o GAP verificar se o mesmo é também uma álgebra de grupo, retornando "true", pois F_2 é um corpo, então $\mathbb{F}_2 D_{16}$ é uma álgebra de grupo.

Em seguida pedimos ao GAP para verificar se $\mathbb{Z}D_{16}$ é uma álgebra de grupo, retornando "false", pois $\mathbb{Z}$ não é corpo, logo $\mathbb{Z}D_{16}$ não é álgebra de grupo.

Agora vamos ver um pouco mais sobre os elementos e operações na álgebra de grupo $\mathbb{Q}D_{16}$ realizados no GAP.

Código 1.2: Operações sobre $\mathbb{Q}D_{16}$

```
gap> KG:=GroupRing(Rationals , DihedralGroup (16) ) ;
<algebra-with-one over Rationals, with 4 generators>
gap> Dimension(KG);   #calculando a dimensao de KG
16
```

```
gap> Zero(KG);     #perguntando ao GAP o elemento zero de KG
<zero> of ...
gap> One(KG);     #perguntando ao GAP a unidade de KG
(1)*<identity> of ...
gap> x:=Random(KG); #x recebe um elemento aleatorio de KG
(-1)*f1+(1/2)*f4+(2)*f1*f2+(1/2)*f1*f3+(-1/2)*f1*f4+
(1)*f2*f4+(2)*f1*f2*f3+(1/2)*f1*f3*f4+(2)*f1*f2*f3*f4
gap> y:=Random(KG);     #y recebe um elemento aleatorio de KG
(1/5)*<identity> of ...+(1/2)*f1+(-2/3)*f4+(1/3)*f1*f2+
(-1/2)*f1*f3+(-3)*f1*f4+(1/3)*f2*f3+(1/2)*f3*f4+
(1)*f1*f2*f3+(-1/2)*f1*f2*f4+(1/2)*f1*f3*f4+
(3/5)*f2*f3*f4+(-1)*f1*f2*f3*f4

gap> Support( x );     #calculando o suporte do elemento x
[ f1, f4, f1*f2, f1*f3, f1*f4, f2*f4, f1*f2*f3, f1*f3*f4,
f1*f2*f3*f4 ]

gap> x+y;      #operando x+y
(1/5)*<identity> of ...+(-1/2)*f1+(-1/6)*f4+(7/3)*f1*f2+
(-7/2)*f1*f4+(1/3)*f2*f3+(1)*f2*f4+(1/2)*f3*f4+
(3)*f1*f2*f3+(-1/2)*f1*f2*f4+(1)*f1*f3*f4+
(3/5)*f2*f3*f4+(1)*f1*f2*f3*f4

gap> x*y;      #operando x*y
(5/3)*<identity> of ...+(-5/12)*f1+(-27/4)*f2+(11/12)*f3+
(49/20)*f4+(127/60)*f1*f2+(19/30)*f1*f3+(31/15)*f1*f4+
(-347/60)*f2*f3+(-52/15)*f2*f4+(-23/6)*f3*f4+
(-47/30)*f1*f2*f3+(-1/5)*f1*f2*f4+
(113/60)*f1*f3*f4+(19/12)*f2*f3*f4+
(-16/5)*f1*f2*f3*f4
```

Nesse trabalho abordamos as álgebras de grupo semissimples, por isso usaremos o pacote Wedderga para realizar os cálculos necessários no GAP. Para entender esse conceito, primeiro precisamos entender o que é um módulo semissimples e o que é um anel semissimples. Para entender estes, precisamos do conceito de um R-módulo.

Definição 1.1.3. *Seja R um anel com unidade* 1*. Diz-se que um conjunto não vazio M é um* ***módulo à esquerda*** *sobre R (ou um R-**módulo à esquerda**) se M é um grupo abeliano em relação a uma operação, que indicaremos por $+$, e está definida uma lei de composição externa que a cada par $(\alpha, m) \in R \times M$ associa um elemento $\alpha m \in M$ tal que, para todos α_1, $\alpha_2 \in R$ e todos m_1, $m_2 \in M$, se verificam:*

- $\alpha_1(\alpha_2 m) = (\alpha_1\alpha_2)m$;
- $\alpha_1(m_1 + m_2) = \alpha_1 m_1 + \alpha_1 m_2$;
- $(\alpha_1 + \alpha_2)m_1 = \alpha_1 m_1 + \alpha_2 m_1$;
- $1m_1 = m_1$

Observação: De forma análoga definimos R-módulo à direita, considerando a multiplicação à direita por elementos do anel.

A partir de agora, chamaremos de R-módulo, ou simplesmente módulo, os R-módulos à esquerda.

Definição 1.1.4. *Seja M um R-módulo. Um subconjunto $N \subset M$ diz-se um R-**submódulo de** M, ou simplesmente, um* ***submódulo*** *se:*

(i) N é um subgrupo aditivo de M.

(ii) N é fechado em relação à multiplicação por escalares, isto é, para todo $r \in R$ e todo $n \in N$, tem-se $rn \in N$.

Definição 1.1.5. *Um R-módulo M é chamado* ***semissimples*** *se todo submódulo de M é um somando direto.*

Definição 1.1.6. *Um R-módulo é dito* ***simples*** *se não possui submódulos próprios não triviais.*

Utilizando a multiplicação num anel R podemos considerá-lo como um R-módulo à esquerda, denotando-o por ${}_RR$ (ou um R-módulo à direita, com a notação R_R).

Definição 1.1.7. *Um **anel** R é chamado* ***semissimples*** *se o R-módulo ${}_RR$ é semissimples.*

Para entender melhor tal definição, vejamos alguns exemplos:

- Um módulo simples M é semissimples, pois $M = M \oplus \{0\}$.
- Corpos e anéis de divisão são anéis semissimples.
- Seja R um anel semissimples. Então o anel $M_n(R)$ das matrizes $n \times n$ sobre R é semissimples.

Além disso, podemos verificar mais rapidamente quando um anel é semissimples simplesmente utilizando o Teorema de Wedderburn-Artin. Para compreender melhor este teorema, primeiro temos o conceito de álgebra.

Definição 1.1.8. *Seja R um anel comutativo. Um R-módulo A é dito uma R-**álgebra** (ou simplesmente **álgebra**) se existe uma multiplicação, definida em A, tal que, com a adição dada em A e sua multiplicação, A é um anel tal que a seguinte condição se verifica:*

$$r(ab) = (ra)b = a(rb),$$

para todo $r \in R$ e todos $a, b \in A$.

Seja D um anel de divisão. O anel $M_n(D)$ das matrizes $n \times n$ sobre D tem estrutura de álgebra.

Teorema 1.1.9. *(Wedderburn-Artin) Um anel R **é semissimples** se, e somente se, R é isomorfo a uma soma direta de álgebras de matrizes sobre anéis de divisão, ou seja,*

$$R \simeq M_{n_1}(D_1) \oplus M_{n_2}(D_2) \oplus \cdots \oplus M_{n_t}(D_t),$$

com $D_1, D_2, \ldots, D_t$ anéis de divisão e $n_1, n_2, ..., n_t$ inteiros positivos.

Note que o Teorema de Wedderburn-Artin apresentado acima se refere a anéis em geral. Para anéis de grupo, uma forma mais direta para se determinar a semissimplicidade é dada pelo Teorema de Maschke.

Teorema 1.1.10. *(Teorema de Maschke) Seja G um grupo. Então o anel de grupo RG* ***é semissimples*** *se, e somente se, as seguintes condições são satisfeitas:*

(i) R é um anel semissimples;

(ii) G é finito;

(iii) $|G|$ é invertível em R.

Segue do Teorema de Maschke o seguinte Corolário que classifica quando uma álgebra de grupo é semissimples.

Corolário 1.1.11. *Seja G um grupo finito e seja K um corpo. Então KG* ***é semissimples*** *se, e somente se, $char(K) \nmid |G|$.*

Utilizando o Teorema de Maschke e o Corolário 1.1.11 podemos reescrever o Teorema de Wedderburn-Artin para anéis de grupo.

Teorema 1.1.12. *(Teorema de Wedderburn-Artin para Álgebras de Grupo)*

Seja G um ***grupo finito*** *e seja K um corpo tal que $char(K) \nmid |G|$. Então*

1. *KG é uma soma direta de um número finito de ideais bilaterais $\{B_i\}_{1 \leqslant i \leqslant r}$, as componentes simples de KG. Cada B_i é um anel simples.*

2. *Qualquer ideal bilateral de KG é um somando direto de alguns dos membros da família $\{B_i\}_{1 \leqslant i \leqslant r}$.*

3. *Cada componente simples B_i é isomorfa a um anel de matrizes completo da forma $M_{n_i}(D_i)$, com D_i é um anel de divisão contendo uma cópia isomorfa de K no seu centro, e o isomorfismo*

$$KG \overset{\phi}{\simeq} \oplus_{i=1}^{r} M_{n_i}(D_i)$$

é um isomorfismo de K-álgebras.

4. *Em cada anel de matrizes completo $M_{n_i}(D_i)$, o conjunto*

$$I_j = \left\{ \begin{bmatrix} 0 & \dots & x_1 & \dots & 0 \\ 0 & \dots & x_2 & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & x_{n_i} & \dots & 0 \end{bmatrix} : x_1, x_2, ..., x_{n_i} \in D_i \right\} \simeq D_i^{n_i}$$

é um ideal à esquerda minimal.

Dado $x \in KG$, considere $\phi(x) = (\alpha_1, ..., \alpha_r) \in \oplus_{i=1}^{r} M_{n_i}(D_i)$ e defina o produto de x por um elemento $m_j \in I_j$ por $xm_j = \alpha_j m_j$. Com esta definição, I_j se torna um KG-módulo simples.

5. *$I_j \not\simeq I_k$, se $j \neq k$.*

6. *Qualquer KG-módulo simples é isomorfo a algum I_j, $1 \leqslant j \leqslant r$.*

1.2 Códigos

O material dessa seção se refere aos Capítulos 1, 5, 6 do livro [8] e ao Capítulo 1 da tese [17].

Definição 1.2.1. *Um **código corretor de erros** é uma maneira de adicionar dados a mais em cada informação que irá ser transmitida ou armazenada, para que seja possível, ao recuperar a informação, detectar e corrigir os erros.*

Ao longo deste trabalho apresentamos alguns códigos corretores de erros sempre tentando encontrar códigos melhores.

Um exemplo de código corretor de erros é a Língua Portuguesa. O comprimento desse código é o número de letras da maior palavra desta Língua, neste caso, pneumoultramicroscopicossilicovulcanoconiótico. Logo, o comprimento da Língua Portuguesa é 46. Tal código é corretor e detector de erros, pois ao receber a palavra "bipicleta", como esta não pertence à Língua Portuguesa, podemos corrigí-la pela palavra mais próxima

pertencente a Língua, que seria "bicicleta". Porém esse código não é muito eficiente, uma vez que podem ocorrer erros que não sejam possíveis de detecção e correção. Como é o caso da palavra "torta", que não seria corrigida caso o erro nos levasse à palavra "porta" ou "corta", por exemplo.

Para construir um código, é necessário um conjunto finito A de caracteres, que é chamado **alfabeto**. O número de elementos de um alfabeto A será denotado por $|A| = q$. Assim, "um **código corretor de erros** é um subconjunto próprio qualquer de A^n, para algum número natural n"([8], p.4), com n o **comprimento do código**, isto é, o comprimento da maior palavra desse código.

Para entender melhor o que seria a distância entre duas palavras, vejamos a definição de distância de Hamming.

Definição 1.2.2. *Seja A um alfabeto e sejam $a, b \in A^n$, a **distância de Hammming** entre a e b é dada por:*

$$d(a,b) = |\{i; a_i \neq b_i, 1 \leqslant i \leqslant n\}|.$$

Para entender melhor essa definição, considere $A = \{1,2,3\}$. Em A^2, temos:

$$d(12,13) = 1, \qquad d(22,13) = 2, \qquad d(12,12) = 0.$$

Definição 1.2.3. *Considere um código C. Chamamos de **distância mínima** de C o número*

$$d = min\{d(a,b) \ / \ a, \ b \in C \ e \ a \neq b\}.$$

Teorema 1.2.4. *Considere um código C cuja distância mínima é d. Assim, C pode detectar no máximo $d-1$ erros e corrigir até $\kappa = \left[\frac{d-1}{2}\right]$ destes erros.*

Definição 1.2.5. *Considere um alfabeto A e um número $n \in \mathbb{N}$. Uma função $F : A^n \to A^n$ é uma **isometria** de A^n se F preserva as distâncias de Hamming, isto é,*

$$d(F(x), F(y)) = d(x,y), \ \ para \ todos \ x, y \in A^n.$$

Definição 1.2.6. *Dois códigos C e $C' \subset A^n$ são **equivalentes** se existir uma isometria $F : A^n \to A^n$ tal que $F(C) = C'$.*

1.2.1 Códigos Lineares

Considere um corpo finito K com q elementos ($q = p^m$) que compõem um alfabeto. Assim, para cada $n \in \mathbb{N}$, temos um K- espaço vetorial K^n de dimensão n.

Definição 1.2.7. *Dizemos que um código C é um **código linear** quando for um subespaço próprio de K^n.*

Logo, todo código linear é um subespaço vetorial de dimensão finita. Para entender melhor esses códigos, considere um código C de dimensão k e uma de suas bases $v_1, v_2, ..., v_k$. Assim, toda palavra do código C pode ser escrita de maneira única como combinação linear dos vetores da base.

$$\alpha_1 v_1 + \alpha_2 v_2 + \cdots + \alpha_k v_k, \text{ com } \alpha_i,\ i = 1, ..., k, \text{ elementos de } K.$$

Assim, a quantidade de palavras do código é $M = |C| = q^k$ e $dim_K C = k = log_q q^k = log_q M$.

Definição 1.2.8. *Seja $x = (x_1, ..., x_n) \in K^n$, chamamos **peso de** x o número*

$$\omega(x) := |\{i \,/\, i \in \mathbb{Z} \text{ e } x_i \neq 0\}|.$$

Note que

$$\omega(x) = d(x, 0), \text{ com } d \text{ a distância de Hamming}.$$

A partir da definição do peso de um elemento de K^n, conseguimos definir o peso de um código.

Definição 1.2.9. *O* ***peso de um código linear*** *C é dado por*

$$\omega(C) := min\{\omega(x) \, / x \in C \backslash \{0\}\}.$$

Em suma, as definições de distância mínima de um código e peso de um código são a mesma, como mostramos na Proposição 1.2.10.

Proposição 1.2.10. *Considere um código linear $C \subset K^n$ com distância mínima d. Assim,*

1. *Para todos $x, y \in K^n$, temos $d(x, y) = \omega(x - y)$;*
2. *$d = \omega(C)$.*

Agora considere um corpo finito K com q elementos e um código linear $C \subset K^n$. Os **parâmetros do código linear** C são a terna (n, k, d), com n o comprimento do código C, k a dimensão de C sobre K e d a distância mínima do código C (que também é o peso do código C). Além disso, sabemos que o número de palavras no código C é q^k.

Considere uma base $\mathcal{B} = \{v_1, .., v_k\}$ do código C. A **matriz geradora do código** é a matriz cujas linhas são os vetores de $v_i = (v_{i1}, ..., v_{1n}),$ com $i = 1, ..., k$.

$$\mathcal{G} = \begin{pmatrix} v_1 \\ \vdots \\ v_k \end{pmatrix} = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ \vdots & \vdots & & \vdots \\ v_{k1} & v_{k2} & \dots & v_{kn} \end{pmatrix}.$$

Uma matriz geradora não é única, pois depende da escolha da base $\mathcal{B}$. Assim, da mesma maneira que nos espaços vetoriais, é possível obter outra matriz geradora $\mathcal{G}'$ para um mesmo código C a partir das seguintes operações com a matriz geradora $\mathcal{G}$:

- Permutação de duas linhas.
- Multiplicação de uma linha por um escalar não nulo.
- Adição de um múltiplo escalar de uma linha a outras.

Definição 1.2.11. *Uma matriz geradora $\mathcal{G}$ de um código C está na* ***forma padrão*** *se*

$$\mathcal{G} = (Id_k|A),$$

com Id_k a matriz identidade $k \times k$ e A uma matriz $k \times (n-k)$.

Apenas com as operações definidas acima nem sempre será possível encontrar uma matriz geradora de um código na forma padrão. Por exemplo, considere o código sobre $\mathbb{F}_2^5$ cuja matriz geradora é

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

Essa matriz geradora não pode ser colocada na forma padrão utilizando somente as operações sobre linhas apresentadas anteriormente. Porém, ao efetuar permutações das colunas de $\mathcal{G}$, podemos obter a seguinte matriz

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

que é a matriz geradora na forma padrão de um código C' equivalente ao código C.

Assim, podemos obter uma matriz geradora $\mathcal{G}'$ de um código C', equivalente ao código C, efetuando também as seguintes operações sobre a matriz $\mathcal{G}$ geradora do código C:

- Permutação de duas colunas,
- Multiplicação de uma coluna por um escalar não nulo.

Note que permutar duas colunas na matriz geradora equivale a permutar a ordem dos vetores respectivos na base do código.

Teorema 1.2.12. *Seja um código C, existe um código equivalente C' que possui uma matriz geradora na forma padrão.*

1.2.2 Códigos Duais

Para entender os códigos duais, primeiro precisamos definir a operação do produto interno. Considere $a = (a_1, ..., a_n)$ e $b = (b_1, ..., b_n) \in K^n$. O produto interno de a e b é dado por

$$\langle a, b \rangle \ = \ a_1 b_1 + \cdots + a_n b_n.$$

Note que tal operação funciona da mesma maneira que o produto interno usual, logo também possui as mesmas propriedades, isto é, possui simetria

$$\langle a, b \rangle \ = \ \langle b, a \rangle \text{ para todos } a,\ b \in K^n$$

e bilinearidade

$$\langle a + \alpha c, b \rangle \ = \ \langle a, b \rangle \ + \ \alpha \langle c, b \rangle \text{ para todos } a,\ b, \in K^n \text{ e } \alpha \in K.$$

A partir do conceito de produto interno, podemos contruir o conjunto

$$C^{\perp} = \{a \in K^n \ / \ \langle a, b \rangle = 0, \text{ para todo } b \in C\}.$$

Daí, temos o seguinte lema.

Lema 1.2.13. *Seja $C \subset K^n$ um código linear com matriz geradora G. Então*

1. *$C^{\perp}$ é um subespaço vetorial de K^n;*
2. *$x \in C^{\perp}$ se, e somente se, $\mathcal{G} x^t = 0$.*

Proposição 1.2.14. *Seja um código linear $C \subset K^n$ de dimensão k que possui uma matriz geradora na forma padrão $\mathcal{G} = (Id_k | A)$. Então*

1. *$dim\, C^{\perp} = n - k$;*
2. *$\mathcal{H} = (-A^t | Id_{n-k})$ é uma matriz geradora do código dual $C^{\perp}$.*

Proposição 1.2.15. *Seja um código linear C e seja $\mathcal{H}$ a matriz geradora de $C^{\perp}$. Então*

$$v \in C \text{ se, e somente se, } \mathcal{H}v^t = 0.$$

A partir desta proposição, temos a Definição 1.2.16.

Definição 1.2.16. *A matriz $\mathcal{H}$ geradora do código linear $C^{\perp}$ é chamada* ***matriz teste de paridade de*** *C.*

Além disso, podemos definir a síndrome de um código.

Definição 1.2.17. *Considere um código C com matriz teste de paridade $\mathcal{H}$. Seja $v \in K^n$, o vetor $\mathcal{H}v^t$ é chamado de* ***síndrome de v****.*

A seguir temos um corolário que estabelece uma cota para os parâmetros de um código linear.

Corolário 1.2.18. *(Cota de Singleton) Para um código linear C, os parâmetros (n, k, d) satisfazem a desigualdade*

$$d \leqslant n - k + 1.$$

Um código que vale a igualdade $d = n - k + 1$ é chamado de MDS (Maximum Distance Separable).

1.2.3 Códigos Cíclicos

Definição 1.2.19. *Um código linear $C \subset K^n$ é chamado um* ***código cíclico*** *se*

$$\text{para todo } x = (x_0, ..., x_{n-1}) \in C, \text{ temos } (x_{n-1}, x_0, ..., x_{n-2}) \in C.$$

Todo código linear pode ser realizado no anel quociente $\dfrac{K[X]}{\langle X^n - 1 \rangle}$ através do se-

guinte isomorfismo de K-álgebras,

$$\begin{array}{rccc} \varphi: & K^n & \to & \dfrac{K[X]}{\langle X^n-1\rangle} \\ & (x_0,...,x_{n-1}) & \mapsto & [x_0+x_1X+...+a_{n-1}X^{n-1}]. \end{array}$$

Assim, um código linear cíclico $C \subset K^n$ se, e somente se, $\varphi(C)$ é um ideal de $\dfrac{K[X]}{\langle X^n-1\rangle}$. Isso significa que, ao estudar os ideais de $\dfrac{K[X]}{\langle X^n-1\rangle}$, estamos estudando os códigos cíclicos de comprimento n sobre K. Assim, $(x_0,...,x_{n-1})$ e $[x_0+x_1X+...+x_{n-1}X^{n-1}]$ são duas maneiras de nos referirmos a uma palavra de um código cíclico contido em K^n.

1.2.4 Códigos de grupo

Denote por $E=\{e_1,...,e_n\}$ a base canônica de K^n.

Definição 1.2.20. *Seja um grupo G de ordem n e um código linear $C \subseteq K^n$. Dizemos que o código C é um G-**código à esquerda (à direita)** se existe uma bijeção ϕ tal que a extensão de $\phi: E \to G$ por linearidade a um isomorfismo de K-espaços vetoriais, dada por $\phi: K^n \to KG$, implica que $\phi(C)$ é um ideal bilateral de KG.*

*Se um código linear C é um G-código (à esquerda, à direita) para algum grupo finito G, então C é um **código de grupo (à esquerda, à direita)**.*

Para as definições a seguir, consideremos a ação natural do grupo simétrico S_n sobre K^n dada por

$$\sigma(x_1,x_2,...,x_n)=(x_{\sigma^{-1}(1)},x_{\sigma^{-1}(2)},...,x_{\sigma^{-1}(n)}), \text{ para cada } (x_1,x_2,...,x_n)\in K^n.$$

Definição 1.2.21. *Dois códigos $C_1, C_2 \subset K^n$ são ditos **equivalentes por permutação** se existe uma permutação $\sigma \in S_n$ tal que $C_2=\sigma(C_1)$.*

Daí temos a definição.

Definição 1.2.22. *Seja um grupo G de ordem n e uma bijeção $\phi : E \to G$. Os* ***G-códigos (à esquerda)*** *sobre K são os códigos lineares contidos em K^n equivalentes por permutação a algum código $\phi^{-1}(I)$, com I um ideal bilateral (à esquerda) de KG.*

Proposição 1.2.23 (Proposição 1.26, [17])**.** *Seja G um grupo de ordem n. Então $C \subseteq K^n$ é G-código (à esquerda) se, e somente se, $C^{\perp}$ é um G-código (à esquerda).*

Definição 1.2.24. *Um código de grupo C que é um A-código, para algum grupo abeliano A, é dito um* ***código de grupo abeliano****.*

Considere um código $C \subset K^n$. O ***grupo de automorfismos permutacionais*** *de C é dado por*

$$PAut(C) = \{\sigma \in S_n \,/\, \sigma(C) = C\}.$$

Capítulo 2

Resultados de códigos de grupo

Esse capítulo aborda alguns resultados apresentados em [2] e [17]. O objetivo principal é apresentar códigos de grupo abelianos e não abelianos. Em relação às funções do GAP, optamos por deixá-las da mesma forma como foram testadas no mesmo. Assim, como o GAP não reconhece acentos, as funções e resultados foram mantidos nesse trabalho sem acentuação.

2.1 Grupos decomponíveis

O primeiro teorema aqui apresentado nos fornece outra maneira de verificar se um código C é um G-código utilizando o grupo de permutações S_n e o grupo de automorfismos permutacionais PAut(C).

Teorema 2.1.1. *[2, Teorema 1.2] Sejam C um código linear de comprimento n sobre um corpo F e G um grupo finito de ordem n.*
(1) C é um G-código à esquerda se, e somente se, G é isomorfo a um subgrupo transitivo de S_n contido em PAut(C).
(2) C é um G-código se, e somente se, G é isomorfo a um subgrupo transitivo H de S_n tal que $H \cup C_{S_n}(H) \subseteq PAut(C)$ (com C_{S_n}(H) denotando o centralizador de H em S_n).

Deste teorema temos o seguinte lema.

Lema 2.1.2. *Um código C de comprimento n é um código abeliano se, e somente se, existe um subgrupo abeliano transitivo de S_n que está contido em PAut(C).*

Demonstração. Segue direto do Teorema 2.1.1.(1) e da definição de código abeliano. □

Considerando dois subconjuntos quaisquer A e B de um grupo G, denotamos por AB o conjunto de todos os produtos ab com $a \in A$ e $b \in B$.

Definição 2.1.3. *Um grupo G é dito* ***decomponível*** *em subgrupos abelianos (ou simplesmente* ***decomponível****) se existem subgrupos abelianos A e B de G tais que $G = AB$.*

Teorema 2.1.4. *[2, Teorema 3.1] Seja G um grupo finito. Se G=AB, com A e B abelianos, então todo G-código é um código de grupo abeliano.*

Esse teorema é um grande auxílio para encontrar códigos de grupo abelianos. Por isso, a seguir veremos alguns resultados relacionados à decomposição de grupos. Primeiramente, vamos lembrar, no Lema 2.1.5, um resultado importante da teoria de grupos que vamos utilizar.

Lema 2.1.5. *Se A, B são dois subgrupos de um grupo G, então*

$$|AB| = \frac{|A||B|}{|A \cap B|}$$

Como consequência do Lema 2.1.5, temos o primeiro resultado sobre decomposição abeliana de grupos.

Lema 2.1.6. *Seja G um grupo com dois subgrupos abelianos A e B tais que $|G| = |A||B|$ e $|A|$, $|B|$ são coprimos. Então G tem uma decomposição abeliana.*

Demonstração. Segue direto da definição de decomposição abeliana e do Lema 2.1.5, uma vez que, como $|A|$ e $|B|$ são coprimos, $|A \cap B| = 1$. □

Corolário 2.1.7. *Se G é um p-grupo não abeliano e $A \leqslant G$, A abeliano com $[G : A] = p$, então G é decomponível.*

Demonstração. Seja G um p-grupo. Daí $|G| = p^n$. Como $[G : A] = p$ e p é o menor primo que divide a ordem de G, então $A \lhd G$. Como $|G/A| = p$, G/A é cíclico gerado por Ag, para algum $g \in G \setminus A$. Logo, $G = A \cup Ag \cup Ag^2 \cup ... \cup Ag^{p-1}$. Portanto, $G = A\langle g \rangle$, isto é, G é decomponível. □

Lema 2.1.8. *Seja G um grupo e N um subgrupo normal abeliano tal que G/N é um grupo cíclico. Então G tem uma decomposição abeliana.*

Demonstração. A demonstração é semelhante a do Corolário anterior. □

Proposição 2.1.9. *[17, Proposição 2.8] Seja G um grupo de ordem $p^i q^j$ com p, q primos, $p \neq q$ e $0<i,j<3$. Então G tem uma decomposição abeliana.*

Demonstração. Sejam A um p-subgrupo de Sylow e B um q-subgrupo de Sylow de G. Então $|A| = p^i$ e $|B| = q^j$ são coprimos e, pelo Lema 2.1.6, G tem uma decomposição abeliana. □

Proposição 2.1.10. *[17, Proposição 2.9] Para qualquer primo p, todo grupo de ordem p^3 e p^4 tem uma decomposição abeliana.*

Demonstração. Seja G um grupo tal que $|G| = p^4$. Suponha G não abeliano. Então $p \leqslant |Z(G)| \leqslant p^3$.

Se $|Z(G)| = p^3$, então $|G/Z(G)| = p$, daí $G/Z(G)$ é cíclico. Assim, temos que G é abeliano, o que é uma contradição. Logo, $Z(G) \neq p^3$.

Se $|Z(G)| = p^2$, então para qualquer $a \notin Z(G)$, o grupo $A = \langle a, Z(G) \rangle$ é abeliano. Pela classificação dos grupos abelianos, ou $G/Z(G)$ é cíclico gerado por $aZ(G)$ ou $G/Z(G)$ é abeliano, não cíclico gerado por $aZ(G)$ e $bZ(G)$. Pelo mesmo argumento usado anteriormente, se $G/Z(G)$ for cíclico, G é abeliano, o que é uma contradição. Logo, $G/Z(G)$ é gerado por $aZ(G)$ e $bZ(G)$. Tomando $A = \langle a, Z(G) \rangle$ e $B = \langle b, Z(G) \rangle$, temos A e B subgrupos abelianos, tais que $G = AB$. Portanto, G tem decomposição abeliana.

Se $|Z(G)| = p$, então $|G/Z(G)| = p^3$. Se $G/Z(G)$ tem um elemento $aZ(G)$ de ordem p^2, então analogamente ao passo que fizemos acima, existe um subgrupo A de

ordem p^3 abeliano pertencente a G. Daí, pelo Lema 2.1.8, como A é normal abeliano em G tal que G/A é cíclico ($[G : A] = p$), então G tem decomposição em abelianos.

Resta provar o caso em que $G/Z(G)$ não tem elemento de ordem p^2 ($G/Z(G)$ não pode ter elementos de ordem p^3, pois ele seria cíclico, logo G seria abeliano), isto é, todos os elementos de $G/Z(G)$ tem ordem p ou 1, no caso do elemento neutro. Assim, para todo $gZ(G) \in G/Z(G)$, $(gZ(G))^p = e$.

Seja $cZ(G) \in Z(G/Z(G))$ tal que $cZ(G) \neq Z(G)$, então $c \notin Z(G)$. Agora $[x, c] \in Z(G)$ para todo elemento $x \in G$. Tome $C = \langle c, Z(G)\rangle$, daí $|C| = p^2$ e $C \lhd G$. Assim, G/C é um grupo elementar abeliano de ordem p^2. Sejam $a, b \in G$ tais que aC, bC geram G/C. Como $[a, c] = z \in Z(G)$ e $[b, c] = w \in Z(G)$, se $[a, c]$ ou $[b, c] = e$ (suponhamos $[a, c] = z = e$), então $\langle a, C\rangle$ é normal abeliano de ordem p^3 e $G/\langle a, C\rangle$ é cíclico. Pelo Lema 2.1.8, G tem decomposição abeliana.

Caso $[a, c] \neq e$ e $[b, c] \neq e$, como $Z(G)$ é um grupo cíclico gerado por algum elemento não trivial, seja $w^{-1} = z^k$, para algum k tal que $0 \leqslant k < p$. Pela igualdade $[xy, u] = [x, u]^y[y, u]$ e $[a, c], [b, c] \in Z(G)$, obtemos:

$$[a^k b, c] = [a^k, c][b, c] = [a, c]^k[b, c] = z^k w = e.$$

Novamente o subgrupo $\langle a^k b, c, Z(G)\rangle$ tem ordem p^3 e é normal abeliano, pois $G/\langle a^k b, c, Z(G)\rangle$ é cíclico de ordem p. Pelo Lema 2.1.8 decorre que G tem decomposição abeliana e isto completa a prova para $|G| = p^4$.

Observe que todo grupo de ordem p^3 é imagem homomorfa de um grupo de ordem p^4, por exemplo, o grupo $G \times \langle a\rangle$, com $\langle a\rangle$ um grupo cíclico de ordem p. Logo, tendo provado o enunciado para p^4, a demonstração para a ordem p^3 é direta. □

Corolário 2.1.11. *Seja G um grupo de ordem $p^i q^j$, com p,q primos tais que $0 < i, j < 3$. Então G tem uma decomposição abeliana*

Demonstração. Pela Proposição 2.1.9, o resultado vale para $p \neq q$. Se $p = q$, então $|G| = p^{i+j}$, com $i + j$ valendo 2, 3 ou 4. Para o caso $i + j = 2$, todo grupo de ordem p^2, para

p um número primo, é abeliano. Os demais casos, $i + j = 3$ ou $i + j = 4$, decorrem da Proposição 2.1.10. □

De acordo com os resultados apresentados até o momento, podemos obter o Corolário 2.1.12.

Corolário 2.1.12. *Para $n \in \{1, ..., 23\}$ todo grupo de ordem n tem uma decomposição abeliana.*

Demonstração. Grupos com ordem $n \in \{1, 2, 3, 4, 5, 7, 9, 11, 13, 15, 17, 19, 23\}$ são abelianos. Os grupos com ordem $n \in \{6, 8, 10, 12, 14, 16, 18, 20, 21, 22\}$ têm decomposição abeliana, pois satisfazem o Corolário 2.1.11. □

Proposição 2.1.13. *[17, Proposição 2.12] O grupo simétrico S_4 não tem decomposição abeliana.*

Demonstração. Os subgrupos abelianos de S_4 tem ordem $1, 2, 3$ ou 4. Fazendo os possíveis produtos de subgrupos abelianos de S_4, o máximo que poderíamos obter é um subgrupo de ordem 16. Logo, S_4 não tem decomposição abeliana. □

Essa prova também pode ser realizada usando o GAP, como mostramos a seguir.

A seguinte função criada por [17] no GAP retorna "true" se foi encontrada uma decomposição Abeliana e "false" caso não seja encontrada tal decomposição.

Código 2.1: Função que verifica se o grupo é decomponível

```
1  HasAbelianDecomposition:=function(G)
2  local lat, A, x, xx, y, z, n, flag;
3  n:=Size(G);
4  lat:=LatticeSubgroups(G);
5  #Calcula o reticulado de todo subgrupo de G
6  A:=Filtered(ConjugacyClassesSubgroups(lat),
7  x->IsAbelian(Representative(x)));
8  #A eh a lista de classes de conjugacao de subgrupos Abelianos
9  flag:=0;
10 for xx in A do x:=Representative(xx);
11     #tome qualquer representacao de uma dada classe
12     for y in A do for z in AsList(y) do
```

```
13          #teste todos os subgrupos abelianos em G
14          if Size(x)*Size(z)/Size(Intersection(x,z))=n then
15              return true;
16          fi;
17      od; od;
18  od;
19  return false;
20  end;
```

Assim, para descobrir se o grupo S_4 tem decomposição abeliana basta fazer no GAP

```
1  HasAbelianDecomposition(SymmetricGroup(4));
```

O resultado da função será "false".

Note que S_4 não é o único grupo de ordem 24 que não possui decomposição abeliana.

Pelo seguinte código obtemos todos os grupos de ordem 24.

```
1  gap> l:=List(AllSmallGroups(Size,24), StructureDescription);
2  [ "C3 : C8", "C24", "SL(2,3)", "C3 : Q8", "C4 x S3", "D24",
3  "C2 x (C3 : C4)", "(C6 x C2) : C2", "C12 x C2", "C3 x D8",
4  "C3 x Q8", "S4", "C2 x A4", "C2 x C2 x S3", "C6 x C2 x C2" ]
```

No passo seguinte verificamos que existem 2 grupos de ordem 24 que não tem decomposição abeliana, S_4 e $SL(2,3)$.

```
1  gap> for i in [1..Length(l)] do
2       if not (HasAbelianDecomposition(SmallGroup(24,i))) then
3           Print(l[i], " nao tem decomposicao abeliana. \n");
4       fi;
5   od;
```

E a resposta obtida foi a seguinte

```
1  SL(2,3) nao tem decomposicao abeliana.
2  S4 nao tem decomposicao abeliana.
```

Podemos fazer o mesmo processo para grupos de qualquer ordem e, assim, verificar a estrutura dos grupos que não tem decomposição abeliana.

Em vista dos resultados obtidos para grupos de ordem p^3 e p^4, com p primo, seria interessante buscar a menor ordem de um p-grupo que não admite decomposição em abelianos.

Baseado na construção de [16] citada em [17] conseguimos encontrar p-grupos finitos que não tem decomposição abeliana. Para um p primo fixo, seja $F(n)$ uma função definida pela seguinte condição: $p^{F(n)}$ é a máxima ordem possível para um p-grupo finito que não possui nenhum subgrupo abeliano de ordem maior que p^n. Segundo [17], pelo Teorema 2 de [16], para algum primo p,

$$F(n) \geqslant \frac{n^2 + 4n - 8}{8}. \tag{2.1}$$

A prova desse teorema inclui uma construção: para um número primo p e um grupo G tal que $|G| = p^m$, dado n tal que G não possui nenhum subgrupo abeliano de ordem excedendo p^n e

$$m = \left[\frac{n^2-1}{8}\right] + \frac{n}{2}, \text{ para } n \text{ par, } m = \left[\frac{n^2-2}{8}\right] + \frac{n+1}{2}, \text{ para } n \text{ ímpar.}$$

Aqui $[\quad]$ denota a função menor inteiro.

Dessa maneira, temos o seguinte teorema.

Teorema 2.1.14. *Dados um primo p e $n \in \mathbb{N}$, se G é um grupo de ordem p^m tal que $m > 2n-1$ e nenhum subgrupo abeliano de G tem ordem superior a p^n, então G não tem decomposição abeliana.*

Demonstração. Suponha $G = AB$, para subgrupos abelianos. Então podemos supor A, B tais que A e B contenham $Z(G)$. Além disso, $Z(G) \neq \langle e \rangle$ e pelo Lema 2.1.5

$$|AB| = \frac{|A||B|}{|A \cap B|} \leqslant \frac{p^n.p^n}{|Z(G)|} \leqslant p^{2n-1} < p^m = |G|,$$

o que é uma contradição. Logo, G não tem decomposição abeliana, concluindo a demons-

tração. □

Vejamos alguns exemplos. Se $n = 13$, então $m = 27 > 2.13 - 1$. Pela construção de [16] retirada de [17] obtemos um grupo G de ordem 2^{27} que não é decomponível.

Porém, os p-grupos não abelianos obtidos utilizando essa teoria possuem uma ordem muito alta. Assim, os resultados apresentados adiante buscam os menores p-grupos sem decomposição abeliana.

Como dito no ínicio do trabalho, estamos procurando códigos não abelianos para analisar seus parâmetros quando comparados aos códigos abelianos. Como em um grupo decomponível todo G-código é um código de grupo abeliano, devemos buscar grupos não decomponíveis para tentar encontrar códigos não abelianos.

O seguinte teorema é um resultado técnico que auxilia na construção de grupos convenientes para os propósitos citados acima.

Teorema 2.1.15. *(Schreier) [7, Teorema 15.1.1] Dado um grupo G com um subgrupo normal N e o quociente $H = G/N$. Se escolhermos representantes de classes $\overline{u}$, com $\overline{u}N \to u \in H$, tomando $\overline{e} = e$, ficam determinados automorfismos e um conjunto fator de N (para $h_1, h_2 \in H, (h_1, h_2) \in N$ é o elemento tal que $\overline{h_1}\,\overline{h_2} = \overline{h_1 h_2}(h_1, h_2)$) que satisfaçam:*

$$\begin{aligned} (a^u)^v &= (u,v)^{-1}(a^{uv})(u,v), \quad a,(u,v) \in N,\, u,v \in H. \\ (uv,w)(u,v)^w &= (u,vw)(v,w), \quad (e,e) = e. \end{aligned} \tag{2.2}$$

Reciprocamente se, para cada $u \in H$, é dado um automorfismo $a \rightleftharpoons a^u$ de N e, para este automorfismo e o conjunto fator, $\{(u,v) \in N, u, v \in H\}$ as condições acima são satisfeitas, então $G = \{\overline{u}a / u \in H, a \in N\}$ com a regra de produto $\overline{u}a.\overline{v}b = \overline{uv}(u,v)a^v b$ é um grupo com subgrupo normal N e $G/N \cong H$.

Teorema 2.1.16. *[17, Teorema 2.13] Para todo primo ímpar p, existe um grupo de ordem p^5 que não tem decomposição abeliana.*

Demonstração. Vamos considerar N um p-grupo abeliano elementar de ordem p^3 , gerado pelos elementos u, z_1 e z_2 e um p-grupo abeliano elementar H de ordem p^2, com gera-

dores $\overline{x}$ e $\overline{y}$. Isto significa que $N \cong Z_p \times Z_p \times Z_p$ e $H \cong Z_p \times Z_p$. Vamos construir uma extensão G com $N \lhd G$ e $G/N \cong H$ tal que, para x e y pré-imagens de $\overline{x}$ e $\overline{y}$, se cumprem as seguintes relações:

$$\begin{gathered} x^p = y^p = e,\ [x,y] = u,\ [x,u] = z_1,\ [y,u] = z_2 \\ [x,z_1] = [y,z_1] = [x,z_2] = [y,z_2] = e. \end{gathered} \tag{2.3}$$

Aplicando o Teorema 2.1.15 de Schreier, os automorfismos do grupo N requeridos $a \mapsto a^h$ se definem por

$$u^{\overline{x}} = uz_1^{-1},\ u^{\overline{y}} = uz_2^{-1},\ z_1^{\overline{x}} = z_1^{\overline{y}} = z_1,\ z_2^{\overline{x}} = z_2^{\overline{y}} = z_2.$$

O conjunto fator pode ser obtido usando (2.3). O cálculo é direto, mas bastante longo, então incluímos apenas a fórmula resultante.

$$(\overline{x}^k\,\overline{y}^l, \overline{x}^r\overline{y}^s) = u^{-rl} z_1^{lr(l-1)/2} z_2^{rls+rl(l-1)/2}, \quad \text{para todos } k,l,r,s \geqslant 0. \tag{2.4}$$

Portanto, as condições do Teorema de Schreier são satisfeitas e apenas resta verificar (2.4), o que significa que se p for adicionado a qualquer um dos números k, l, r ou s, então o resultado não varia, pois $\overline{x}, \overline{y}, u, z_1, z_2$ são geradores de grupos de ordem p. Como, por hipótese, p é ímpar e G é a extensão descrita anteriormente, primeiro, observamos que $|G| = p^5$, já que $|H| = p^2$ e $|N| = p^3$.

A seguir, provamos que o grupo G não contém subgrupos abelianos de ordem p^4. Suponha que exista um subgrupo A abeliano de ordem p^4, então ele deve ser normal em G já que $[G : A] = p$ e p é o menor primo divisor de $|G|$. Sabemos que $N = G'$, uma vez que $N \subset G'$, pois $u, z_1, z_2 \in G'$, pelas relações 2.3 e $G' \subset N$, já que G/N é abeliano. Além disso, como $|G/A| = p$, o grupo G/A também é abeliano e daí $A \supset G' = N$.

Seja $a \in A \backslash N$. Tomamos, sem perda de generalidade, $a = x^k y^l$, com $(k,l) \not\equiv (0,0)(mod\ p)$. Como A é abeliano, para $a \in A$ e $u \in A$, temos $au = ua$. Daí

$$u = a^{-1}ua = y^{-l}x^{-k}ux^k y^l = uz_1^k z_2^l \neq u,$$

uma contradição. Logo, não existe subgrupo abeliano de ordem p^4 em G.

Suponhamos agora que existam dois subgrupos abelianos A, B de G com $G = AB$. Podemos supor que ambos subgrupos contenham o centro $Z(G) = \langle z_1, z_2 \rangle$ do grupo G, posto que $AZ(G)$ e $BZ(G)$ são subgrupos abelianos com a mesma propriedade. Nesse caso, $|A \cap B| \geqslant |Z(G)| = p^2$ e, pelo Lema 2.1.5, $|AB| \leqslant p^3.p^3/p^2 = p^4 < |G|$, uma contradição. Portanto, G não tem decomposição abeliana. □

Proposição 2.1.17. *[17, Proposição 2.14] Todo grupo de ordem 32 pode ser escrito como um produto de dois grupos abelianos.*

Demonstração. Para essa proposição, temos a prova teórica e a prova computacional, realizada no GAP. Começaremos mostrando a prova teórica apresentada em [17].

Se $|G| = 2^5$ e G não é um grupo abeliano, existem três possíveis casos para seu comprimento nilpotente (denotado por $l(G)$): 2, 3 ou 4. Vamos estudar cada um desses separadamente.

1. Tome $l(G) = 2$.

   Neste caso, a série central superior é

   $$1 = Z_0(G) \leqslant Z_1(G) = Z(G) \leqslant Z_2(G) = G,$$

   isto é, $G/Z(G)$ é abeliano.

   Devemos considerar três possibilidades.

   (a) Se $|Z(G)| = 2$ e $|G/Z(G)| = 16\,(Z(G) = \langle z \rangle,\ z^2 = 1)$, então existem quatro possibilidades para $G/Z(G)$: $C_2 \oplus C_8$, $C_4 \oplus C_4$, $C_2 \oplus C_2 \oplus C_4$ e $C_2 \oplus C_2 \oplus C_2 \oplus C_2$.

      (i) Nos primeiros dois casos, $G/Z(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle$, então $A = \langle a \rangle Z(G)$ e $B = \langle b \rangle Z(G)$ são abelianos e $G = AB$.

      (ii) Se $G/Z(G) = C_2 \oplus C_2 \oplus C_4 = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle \oplus \langle \overline{c} \rangle$, então $[a, c] = z = [b, c]$ implica que $[ab, c] = [a, c]^b[b, c] = z^b z = z^2 = 1$. Logo, sem perda de

generalidade, podemos assumir $[a,c]=1$ tal que $A=\langle a,c\rangle$ e $B=\langle b\rangle Z(G)$ são abelianos e $G=AB$.

(iii) Seja $G/Z(G)=\langle\overline{a}\rangle\oplus\langle\overline{b}\rangle\oplus\langle\overline{c}\rangle\oplus\langle\overline{d}\rangle\cong C_2\oplus C_2\oplus C_2\oplus C_2$.
Se $[a,b]=z=[a,c]$, então $[a,bc]=1$. Logo, sem perda de generalidade, podemos supor $[a,b]=1$.
Se $[a,c]=[b,c]=1$, então $A=\langle a,b,c\rangle$ e $B=\langle d\rangle Z(G)$ são abelianos e $G=AB$.
Se $[c,d]=1$, $A=\langle a,b\rangle Z(G)$ e $B=\langle d,c\rangle$ são abelianos e $G=AB$.
Se $[c,d]=z=[a,c]$ então $[c,ad]=1$, assim $A=\langle a,b\rangle Z(G)$ e $B=\langle c,ad\rangle$ são abelianos e $G=AB$.

(b) Se $|Z(G)|=4$ e $|G/Z(G)|=8$, então existem duas possibilidades para $G/Z(G)$: $C_4\oplus C_2$ e $C_2\oplus C_2\oplus C_2$. O primeiro caso segue como 1(a)(i), então podemos assumir $G/Z(G)\cong C_2\oplus C_2\oplus C_2\cong\langle\overline{a}\rangle\oplus\langle\overline{b}\rangle\oplus\langle\overline{c}\rangle$.

(i) Suponha $Z(G)=\langle z\rangle\cong C_4$.
Se $[a,b]=1$ (ou $[a,c]=1$ ou $[b,c]=1$), então $A=\langle a,b\rangle$ e $B=\langle c\rangle Z(G)$ são abelianos e $G=AB$.
Se dois comutadores, por exemplo $[a,b]=z^2=[a,c]$, então $[a,bc]=1$ e retornamos ao caso anterior.
Se $[a,b]=z$, então $1=[a^2,b]=[a,b]^a[a,b]=z^az=z^2$, o que contradiz o fato que $Z(G)=\langle z\rangle\cong C_4$.

(ii) Vamos supor $Z(G)=C_2\oplus C_2=\langle z_1\rangle\oplus\langle z_2\rangle, z_1^2=z_2^2=1$.
Se $[a,b]=1$ (ou $[a,c]=1$ ou $[b,c]=1$) então $A=\langle a,b\rangle$ e $B=\langle c\rangle Z(G)$ são abelianos e $G=AB$.
Vamos assumir que $[a,b]=z_1, [a,c]=z_2$ e $[b,c]=z_1z_2$. Então

$$\begin{cases} [ab,c] &= [a,c]^b[b,c] &= z_2^bz_1z_2 &= z_1, \\ [ab,b] &= [a,b]^b[b,b] &= z_1. \end{cases}$$

Daí $[ab,bc]=z_1^2=1$ e consequentemente $A=\langle ab,bc\rangle$ e $B=\langle b\rangle Z(G)$ são abelianos e $G=AB$.

(c) Se $|Z(G)| = 8$, então $G/Z(G) \cong C_2 \oplus C_2$ e podemos proceder como em 1(a)(i).

2. Seja $l(G) = 3$.

Deste modo, a série central ascendente é

$$1 = Z_0(G) \leqslant Z_1(G) \leqslant Z_2(G) \leqslant Z_3(G) = G,$$

e a série central descendente é

$$G = G^{(1)} \supset G^{(2)} = [G, G] \supset G^{(3)} = [G, G^{(2)}] \supset G^{(4)} = [G, G^{(3)}] = 1.$$

Daí $G^{(3)} \subseteq Z_1(G)$ e, já que $G/Z_2(G)$ é abeliano, $G^{(2)} \subseteq Z_2(G)$.

Existem duas possibilidades:

a) Se $|G/Z_2(G)| = 8$, então $G/Z_2(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle \cong C_2 \oplus C_4$ ou $G/Z_2(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle \oplus \langle \overline{c} \rangle \cong C_2 \oplus C_2 \oplus C_2$.

Em qualquer caso, $Z_1(G) = \langle z \rangle \cong C_2$ e $Z_2(G)/Z_1(G) = \langle \overline{u} \rangle \cong C_2$.

(i) Vamos considerar $G/Z_2(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle \cong C_2 \oplus C_4$.
Se $[a, u] = 1$ (ou $[b, u] = 1$), então $A = \langle a \rangle Z_2(G)$ e $B = \langle b \rangle$ são abelianos e $G = AB$ (respectivamente $A = \langle a \rangle$ e $B = \langle b \rangle Z_2(G)$).
Se $[a, u] = z = [b, u]$, então $[ab, u] = [a, u]^b [b, u] = z^b z = z^2 = 1$. Segue que $A = \langle ab \rangle Z_2(G)$ e $B = \langle b \rangle$ são abelianos e $G = AB$.

(ii) Vamos assumir agora $G/Z_2(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle \oplus \langle \overline{c} \rangle \cong C_2 \oplus C_2 \oplus C_2$.
Já que $|Z_1(G)| = 2$ e $|Z_2(G)| = 4$, $G^{(2)} = Z_2(G)$ e $G^{(3)} = Z_1(G)$.
Como $[a, u], [b, u]$ e $[c, u] \in Z_1(G)$, podemos assumir, sem perda de generalidade, $[a, u] = 1$.
Se $[b, u] = z = [c, u]$, então $[bc, u] = 1$. Logo, pode-se supor $[a, u] = 1 = [b, u]$ e $[c, u] = z$.

- Se $[a, b] = 1$, então $A = \langle a, b \rangle Z_2(G)$ e $B = \langle c \rangle$ são abelianos e $G = AB$.

- Se $[a,b] = u$, já que $[a^2,b] \in [Z_2(G),b] = 1$, então $1 = [a^2,b] = [a,b]^a[a,b] = u^a u = u^2$. Logo, $o(u) = 2$, isto é, $Z_2(G) \cong C_2 \oplus C_2$. Se $[a,c] = u$, então $[a,cb] = [a,b][a,c]^b = uu^b = u^2 = 1$. Portanto, $A = \langle a, cb, z\rangle$ e $B = \langle b, u\rangle$ são abelianos e $G = AB$.
- Se $[a,b] = z$, podemos assumir, sem perda de generalidade, $[a,c] = u$ ($Z_2(G) = G^{(2)}$). Já que $c^2 \in Z_2(G)$, $[a,c^2] \in [a, Z_2(G)] = 1$, então $1 = [a,c^2] = [a,c][a,c]^c = uu^c$ e $u^c = c^{-1}uc = [c,u^{-1}]u = u[u,c] = uz$, que implica $1 = uu^c = u^2z$., isto é, $Z_2(G) \cong C_4 = \langle u\rangle$ e $u^2 = z$.
  Agora $[b,c] = u = [a,c]$ leva a $[ab,c] = [a,c]^b[b,c] = u^b u = u^2 = z$.
  Além disso, $[a,b] = z$ implica $[ab,b] = [a,b]^b[b,b] = z^b = z$, então $[ab,bc] = 1$ e, portanto, $A = \langle a\rangle Z_2(G)$ e $B = \langle ab, bc\rangle$ são abelianos e $G = AB$.
  O caso $[b,c] = z$ implica $[ac,b] = 1$ e $A = \langle ac, b, z\rangle$, $B = \langle a, u\rangle$ formam uma decomposição abeliana de G. Claramente $[b,c] = 1$ nos leva à decomposição abeliana $G = \langle b,c\rangle\langle a,u\rangle$.

b) Se $|G/Z_2(G)| = 4$, então $|Z_2(G)| = 8$, $G/Z_2(G) = \langle\overline{a}\rangle \oplus \langle\overline{b}\rangle \cong C_2 \oplus C_2$ e $Z_2(G)$ pode ser C_8, $C_4 \oplus C_2$, $C_2 \oplus C_2 \oplus C_2$, Q_8 ou D_4.

(i) Se $Z_2(G) \cong C_8 = \langle v\rangle$, então $Z(G) = \langle v^2\rangle$ ou $Z(G) = \langle v^4\rangle$.

- Suponha $Z(G) = \langle v^2\rangle \cong C_4$. Já que $l(G) = 3, [a,b] = v$ $(a^2 \in Z_2(G))$.
  Se $a^2 \in Z_1(G)$ (ou $b^2 \in Z_1(G)$), então $[a^2,b] = 1 = [a,b]^a[a,b] = v^a v$.
  Logo $v^a = v^{-1}$ e $[a,v] = v^2$.
  Agora $1 = [a,v^2] = [a,v][a,v]^v = v^2v^2 = v^4$, o que contradiz $o(v) = 8$.
  Portanto, $a^2 \notin Z_1(G)$ e $b^2 \notin Z_1(G)$. Já que $a^2 \notin Z_1(G), A = \langle a\rangle Z_1(G)$ é abeliano e $|A| = 16$, então $G = A\langle b\rangle$.
- Se $Z(G) = \langle v^4\rangle \cong C_2$, $[a,v] = 1$ ou v^4.
  Se $[a,v] = 1$ (ou $[b,v] = 1$), então $A = \langle a\rangle Z_2(G)$ é abeliano e, portanto, $G = A\langle b\rangle$.
  Se $[a,v] = v^4$ e $[b,v] = v^4$, então $[ab,v] = 1$ e consequentemente $A = \langle ab\rangle Z_2(G)$ é abeliano e $G = A\langle b\rangle$.

(ii) Se $Z_2(G) \cong C_2 \oplus C_4 = \langle u \rangle \oplus \langle v \rangle$, existem quatro diferentes possibilidades para ser consideradas para $Z(G)$: $\langle u \rangle$, $\langle v^2 \rangle, \langle v \rangle$ e $\langle u \rangle \times \langle v^2 \rangle$ (os casos $Z(G) = \langle uv^2 \rangle$ e $Z(G) = \langle uv \rangle$ comportam-se da mesma forma que $Z(G) = \langle u \rangle$ e $Z(G) = \langle v \rangle$, respectivamente).

- Suponha $Z(G) = \langle u \rangle$. Então $[a, v] = 1$ ou u e $[b, v] = 1$ ou u. Podemos usar o mesmo argumento de 2(b)(i).
- Se $Z(G) = \langle v^2 \rangle$, usando o mesmo argumento de antes, podemos assumir que um dos elementos $[a, u], [b, u]$ é 1 e, da mesma forma, um dos elementos $[a, v], [b, v]$ é 1.
  Se $[a, u] = [a, v] = 1$, então $A = \langle a, u, v \rangle$ é abeliano e $G = A\langle b \rangle$.
  Se $[a, u] = 1 = [b, v]$, então $A = \langle a, u \rangle$ e $B = \langle b, v \rangle$ são abelianos e $G = AB$.
- Suponha $Z(G) = \langle v \rangle$. Então $[a, u] \in Z(G)$ e $1 = [a, u^2] = [a, u]^2$ implica $[a, u] = 1$ ou v^2. De maneira semelhante, obtemos $[b, u] = 1$ ou v^2.
  Agora $[a, u] = v^2 = [b, u]$ implica $[ab, u] = 1$, portanto, sem perda de generalidade, podemos supor $[a, u] = 1$. Assim, $A = \langle a \rangle Z_2(G)$ e $B = \langle b \rangle$ são abelianos e $G = AB$.
- Suponha $Z(G) = \langle u \rangle \times \langle v^2 \rangle$ e $[a, b] = v$.
  Se $[a, v] = 1$ (ou $[b, v] = 1$), então $A = \langle a \rangle Z_2(G)$ é abeliano e $G = A\langle b \rangle$.
  Se $[a, v] = u = [b, v]$ (ou qualquer outro elemento de ordem 2), então $[ab, v] = u^2 = 1$ e podemos utilizar o mesmo argumento.
  Suponha $[a, v] = u, [b, v] = v^2$.
  Se $a^2 \in Z(G)$, $1 = [a^2, b] = [a, b]^a[a, b] = v^a v = a^{-1}vav = vv^{-1}a^{-1}vav = v[v, a]v = v^2u$, o que é uma contradição.
  Se $a^2 \in Z(G)$, $A = \langle a \rangle Z(G)$ é abeliano e, já que, $|A| = 16$, $G = A\langle b \rangle$.

(iii) Vamos assumir agora $Z_2(G) \cong C_2 \oplus C_2 \oplus C_2$. Nesse caso, $Z(G) \cong C_2$ ou $Z(G) \cong C_2 \oplus C_2$.

- Se $Z(G) = \langle z_1 \rangle \oplus \langle z_2 \rangle$ e $Z_2(G) = \langle u \rangle \oplus \langle z_1 \rangle \oplus \langle z_2 \rangle$, então podemos assumir $[a, b] = u$.

  Se $[a, u] = 1$, então $A = \langle a \rangle Z_2(G)$ é abeliano e $G = A\langle b \rangle$.

  Se $[a, u] \neq 1$, $[a^2, b] = [a, b]^a [a, b] = u^a u = a^{-1}u^{-1}au = [a, u] \neq 1$ então $a^2 \notin Z(G)$ e $A = \langle a \rangle Z(G)$ é abeliano. Também, como $|A| = 16$, $G = A\langle b \rangle$.

- Se $Z(G) = \langle z \rangle \cong C_2$, então $Z_2(G) = \langle u \rangle \oplus \langle v \rangle \oplus \langle z \rangle$ e, sem perda de generalidade, podemos supor $[a, b] = u$. Além disso, $[a, u] = 1$ ou z, $[b, u] = 1$ ou z.

  Sem perda de generalidade, podemos assumir $[a, u] = 1$ e, ou $[a, v] = 1$ ou $[b, v] = 1$.

  Se $[a, v] = 1$, então $A = \langle a \rangle Z_2(G)$ é abeliano e $G = A\langle b \rangle$.

  Se $[b, v] = 1$ e $[a, v] = z$, então $A = \langle a, u, z \rangle$ e $B = \langle b, v \rangle$ são abelianos e $G = AB$.

(iv) Se $Z_2(G) \cong Q_8 = \langle u, v / u^4 = 1, u^2 = v^2 = z, u^v = u^1 \rangle$ e $Z(G) = Z(Q_8) = \langle z \rangle$, então, sem perda de generalidade, podemos assumir $[a, b] = u$ ($G^{(2)} = \langle u \rangle$).

Se $[a, u] = z = [b, u]$, então $[ab, u] = 1$, sem perda de generalidade, podemos supor $[a, u] = 1$.

Se $[b, u] = z$, sem perda de generalidade, podemos supor $[b, v] = 1$ (se $[b, v] = z$ implica $[b, uv] = 1$ e uv desempenha o mesmo papel que v). Portanto,

$A = \langle a, u \rangle$ e $B = \langle b, v \rangle$ são abelianos e $G = AB$.

Agora vamos supor $[b, u] = 1$ e $[b, v] = z = [a, v]$. Neste caso, $[ab, v] = 1$ e $A = \langle a, u \rangle$ e $B = \langle ab, v \rangle$ são abelianos e $G = AB$.

(v) Suponha $Z_2(G) \cong D_4 = \langle u, v / u^4 = 1 = v^2, vuv = u^{-1} \rangle$, $Z_1(G) = \langle u^2 = z \rangle$.

Visto que $[a, u]$, $[b, u]$ e $[ab, u] = 1$ ou z e, da mesma forma $[a, v]$, $[b, v]$ e $[ab, v] = 1$ ou z, podemos assumir, sem perda de generalidade, que $[a, u] = 1$ e um dos elementos $[a, v], [b, v]$ é igual a um. Se $[b, v] = 1$, então

$A = \langle a, u \rangle$ e $B = \langle b, v \rangle$ são abelianos e $G = AB$.
Se $[a, u] = 1 = [a, v]$ e $[b, v] = z$, podemos assumir $[b, u] = z$ (caso contrário, $[b, u] = 1$ e novamente $G = AB$, com $A = \langle a, v \rangle$, $B = \langle b, u \rangle$).
Logo, $[b, uv] = 1$ e, em seguida, $G = AB$, com $A = \langle a, u \rangle$ e $B = \langle b, uv \rangle$ abelianos.

(3) Seja $l(G) = 4$.

Deste modo, a série central ascendente é

$$1 = Z_0(G) \leqslant Z_1(G) \leqslant Z_2(G) \leqslant Z_3(G) \leqslant Z_4(G) = G,$$
$$Z_1(G) \cong C_2,\ Z_2(G)/Z_1(G) \cong C_2,\ Z_3(G)/Z_2(G) \cong C_2,$$
$$G/Z_3(G) \cong C_2 \oplus C_2.$$

Note que $G^{(2)} = Z_3(G)$, $G^{(3)} = Z_2(G)$, $G^{(4)} = Z_1(G)$.

Sejam $Z_1(G) = \langle z \rangle$, $Z_2(G) = \langle v \rangle Z_1(G)$, $Z_3(G) = \langle u \rangle Z_2(G)$ e $G/Z_3(G) = \langle \overline{a} \rangle \oplus \langle \overline{b} \rangle$.

Podemos supor, sem perda de generalidade, $[a, b] = u$. Se $[a, v] = z$ e $[b, v] = z$, então $[ab, v] = 1$. Daí podemos assumir, sem perda de generalidade, $[a, b] = u, [a, v] = 1$.

Já que, $[a, v] = 1$, $[a, Z_2(G)] = 1$, então $A = \langle a \rangle Z_2(G)$ é abeliano. Se $a^2 \notin Z_2(G)$, então $|A| = 16$ e $G = AB$ ($B = \langle b \rangle$). Por conseguinte, no que segue, consideramos o caso $a^2 \in Z_2(G) = \langle v, z \rangle = G^{(3)}$.

Dependendo da estrutura de $Z_3(G)$ temos cinco possibilidades:

a) Se $Z_3(G) \cong D_4$, existem duas possibilidades:

$$\begin{aligned} Z_3(G) &= \langle u, v / u^4 = 1 = v^2,\ u^2 = z,\ [u, v] = z \rangle \\ &\text{ou} \\ Z_3(G) &= \langle v, u / v^4 = 1 = u^2, v^2 = z, [u, v] = z \rangle. \end{aligned}$$

Note que, $u^2 \in Z_1(G)$ sempre. Podemos assumir $[a,u] = v$ ou $[b,u] = v$.

Como $a^2 \in Z_2(G)$, temos $[a^2,b] \in Z_1(G)$ e $[a^2,b] = [a,b]^a[a,b] = u^a u = a^{-1}uau = a^{-1}u^{-1}au = [a,u]$. Logo, $[a,u] \in Z_1$. Assim, sem perda de generalidade, $[b,u] = v$. Daí $[b,u^2] = 1 = [b,u][b,u]^u = vv^u$ e $Z_3(G)$ é tipo (2).

Se $b^2 \in Z_2(G)$, então $[a,b^2] \in Z_1(G)$ e $[a,b^2] = [a,b][a,b]^b = uu^b = u^{-1}b^{-1}ub = [u,b] = v^{-1}$. Consequentemente, $b^2 \notin Z_2(G)$ e podemos assumir, sem perda de generalidade, $b^2 = u$. Assim, por um lado, $[a,b^2] = [a,u] \in Z_1(G)$ e, por outro lado, $[a,b^2] = [a,b][a,b]^b = uu^b = v^{-1}$, o que é uma contradição.

b) Seja $Z_3(G) = Q_8 = \langle u,v / u^4 = 1, u^2 = v^2 = z, [v,u] = z\rangle$.

Estamos supondo $[a,b] = u, [a,v] = 1$ e $a^2 \in Z_2(G)$. Logo, $[a^2,b] \in Z_1(G)$, $[a^2,b] = [a,b]^a[a,b] = u^a u = [a,u^{-1}]u^2 = [a,u^{-1}]z$. Assim, $[a,u^{-1}] \in Z_1(G)$, o que implica $[a,u] \in Z_1(G)$ e daí $[a,u] = 1$ ou z.

Dado $Z_3(G) = G^{(2)} = \langle v\rangle$, podemos supor $[b,u] = v$. Se $b^2 \in Z_2(G)$, então $[a,b^2] \in Z_1(G)$ e como antes $[a,b^2] = [a,b][a,b]^b = uu^b = u^2u^{-1}b^{-1}ub = u^2[u,b] = u^2v^{-1} = v$, o que é uma contradição. Assim, $b^2 \notin Z_2(G)$, isto é, $b^2 = u,\ u^{-1}, uv, u^{-1}v$. Posto que $[b,u] = v \neq 1$, podemos concluir $b^2 = uv$ ou $b^2 = u^{-1}v$ e $G = AB$, com $A = \langle a\rangle Z_2(G) = \langle a\rangle\langle v\rangle$ e $B = \langle b\rangle$ abelianos.

c) Seja $Z_3(G) = \langle u\rangle \cong C_8,\ \ v = u^2,\ z = u^4$.

Temos $[a,b] = u, [a,v] = 1$ e $[a,u] \in Z_2(G)$. Assim, $1 = [a,u^2] = [a,u][a,u]^u = [a,u]^2$. Logo, $[a,u] = 1$ ou u^4.

Se $[a,u] = 1$, então $A = \langle a\rangle Z_3(G)$ é abeliano e $G = A\langle b\rangle$.

Logo, $[a,u] = u^4$. Agora $[a,u] = u^4$ implica $u^a = u[u,a] = uu^4 = u^5$.

$$[a^2,b] \in [Z_2(G),G] = Z_1(G),$$

$$[a^2,b] = [a,b]^a[a,b] = u^a u = u^5 u = u^6 \notin Z_1(G),$$

o que é uma contradição.

d) Seja $Z_3(G) \cong C_2 \oplus C_2 \oplus C_2 = \langle u\rangle \oplus \langle v\rangle \oplus \langle z\rangle$.

Como nos casos anteriores $[a,b] = u, [a,v] = 1$. Assim, usando $Z_1(G) =$

$[G, Z_2(G)]$, podemos assumir $[b, v] = z$. Assim, $[a^2, b] \in [Z_2(G), G] = Z_1(G)$, $[a^2, b] = [a, b]^a[a, b] = u^a u = [a, u]$, logo $[a, u] \in Z_1(G)$.

Se $[a, u] = 1$, então $A = \langle a \rangle Z_3(G)$ é abeliano e $G = A\langle b \rangle$.

Se $[a, u] = z$, então, sem perda de generalidade, $[b, u] = v$ ($G^{(3)} = Z_2(G)$). Como $b^2 \in Z_3(G)$ (que é abeliano), $1 = [b^2, u] = [b, u]^b[b, u] = v^b v = [b, v] = z$, o que é uma contradição.

e) Seja $Z_3(G) \cong C_4 \oplus C_2 = \begin{cases} \langle u \rangle \oplus \langle z \rangle, \ u^2 = v \ (1) \\ \langle u \rangle \oplus \langle v \rangle \ , \ u^2 = z \ (2) \\ \langle v \rangle \oplus \langle u \rangle, \ v^2 = z \ (3) \end{cases}$

Como no caso anterior, podemos supor, sem perda de generalidade, $[a, b] = u, [a, v] = 1$ e $a^2 \in Z_2(G)$.

Como $[a, b] = u, [a, v] = 1$ e $a^2 \in Z_2(G)$, temos $[a^2, b] \in [Z_2(G), G] = Z_1(G)$. Agora $[a^2, b] = [a, b]^a[a, b] = u^a u$.

No caso (3), $u = u^{-1}$, $[a^2, b] = [a, u] \in Z_1(G)$.

No caso (2), $[a^2, b] = u^a u = [a, u^{-1}]u^2 = [a, u^{-1}]z \in Z_1(G)$, logo $[a, u^{-1}] \in Z_1(G)$.

Nestes casos, $[a, u] = 1$ ou z e podemos assumir $[b, u] = v, [b, v] = z$.

Se $[a, u] = 1$, então $A = \langle a \rangle Z_3(G)$ e $B = \langle b \rangle$ são abelianos e $G = AB$.

Se $[a, u] = z$, como $[b^2, u] = 1$ ($b^2 \in Z_3(G)$ abeliano) $1 = [b, u]^b[b, u] = v^b v$.

Se $o(v) = 2$, então $1 = b^{-1}vbv = b^{-1}v^{-1}bv = [b, v]$, o que é uma contradição. Consequentemente, $o(v) = 4$ e $Z_3(G)$ é do tipo (3).

Como antes, $b^2 \notin Z_2(G)$ (como vimos em (3(a)) $b^2 \in Z_2(G)$ implica $[b, u] \in Z_1(G)$, já que $u^{-1} = u$). Portanto, $b^2 \in uZ_2$. Então $[a, u] = [a, b^2] = [a, b][a, b]^b = uu^b = ub^{-1}ub = u^{-1}b^{-1}ub = [u, b] = v$, o que é contradição, pois $[a, u] \in Z_1(G)$.

Vamos considerar o caso $[a, u] \notin Z_1(G)$. Como vimos, isso implica que $Z_3(G)$ é do tipo (1). Assim, podemos assumir $[a, u] = v$ e $[b, v] = z$.

Se $[b, u] = 1$, então $G = AB$, com $A = \langle a \rangle Z_2(G)$ e $B = \langle u, b \rangle$ abelianos.

Se $[b,u] = z, [b,vu] = [b,u][b,v]^u = zz^u = z^2 = 1$, então, $G = AB$, com $A = \langle a \rangle Z_2(G)$ e $B = \langle b, uv \rangle$ abelianos.

Se $[b,u] = v$, então $[ab,u] = [a,u]^b[b,u] = v^b v = b^{-1}vbv = v^2 z = 1$. Se considerarmos ab ao invés de b, encontramos a situação do caso anterior (note que $[ab,v] = [b,v]$).

Agora mostraremos a prova computacional realizada no GAP.

Faremos aqui a demonstração utilizando o GAP e a função "HasAbelianDecomposition" apresentada em [17].

Observação: O produto de grupos é indicado pelo sinal:

```
"x".
```

```
 gap> l:=List(AllSmallGroups(Size,32), StructureDescription);
[ "C32", "(C4 x C2) : C4", "C8 x C4", "C8 : C4", "(C8 x C2) : C2", "(C2 x C2 x C2) : C4",
"(C8 : C2) : C2", "C2 . ((C4 x C2) : C2) = (C2 x C2) . (C4 x C2)", "(C8 x C2) : C2",
"Q8 : C4", "(C4 x C4) : C2", "C4 : C8", "C8 : C4", "C8 : C4", "C4 . D8 = C4 . (C4 x C2)",
"C16 x C2", "C16 : C2", "D32", "QD32", "Q32", "C4 x C4 x C2", "C2 x ((C4 x C2) : C2)",
"C2 x (C4 : C4)", "(C4 x C4) : C2", "C4 x D8", "C4 x Q8", "(C2 x C2 x C2 x C2) : C2",
"(C4 x C2 x C2) : C2", "(C2 x Q8) : C2", "(C4 x C2 x C2) : C2", "(C4 x C4) : C2",
"(C2 x C2) . (C2 x C2 x C2)", "(C4 x C4) : C2", "(C4 x C4) : C2", "C4 : Q8", "C8 x C2 x C2",
"C2 x (C8 : C2)", "(C8 x C2) : C2", "C2 x D16", "C2 x QD16", "C2 x Q16", "(C8 x C2) : C2",
"C8 : (C2 x C2)", "(C2 x Q8) : C2", "C4 x C2 x C2 x C2", "C2 x C2 x D8", "C2 x C2 x Q8",
"C2 x ((C4 x C2) : C2)", "(C2 x C2 x C2) : (C2 x C2)", "(C2 x Q8) : C2",
"C2 x C2 x C2 x C2 x C2" ]

gap> cont:=0;
0
gap> for i in [1..Length(l)] do
        if (HasAbelianDecomposition(SmallGroup(32,i))) then
            cont:=cont+1;
        fi;
     od;

gap> Print("Existem ", Length(l), " grupos de ordem 32 e ", cont,
```

```
23  " deles tem decomposicao abeliana. \n");
24  Existem 51 grupos de ordem 32 e 51 deles tem decomposicao abeliana.
```

Logo, todos os grupos de ordem 32 podem ser escritos como produto de dois grupos abelianos. □

Corolário 2.1.18. *O grupo derivado de um grupo de ordem 32 é abeliano.*

Demonstração. Pela Proposição 2.1.17, $G = AB$, com A e B abelianos. Então G/A é abeliano, daí $G' \subset A$. Logo, G' é abeliano. □

Proposição 2.1.19. *Existe um grupo G tal que $Z(G)$ e $G/Z(G)$ são grupos elementares abelianos de ordem 8, mas G não tem uma decomposição abeliana.*

Demonstração. Para construir o grupo G desejado, procedemos como na prova do Teorema 2.1.16. Considere um grupo abeliano 2-elementar N com três geradores z_1, z_2 e z_3 e também um grupo abeliano 2-elementar H com três geradores $\overline{x_1}, \overline{x_2}$ e $\overline{x_3}$. Construimos uma extensão G com $N = Z(G) \lhd G$ e $G/N \cong H$ tal que as pré-imagens x_1, x_2 e x_3 de $\overline{x_1}, \overline{x_2}$ e $\overline{x_3}$ satisfazem as seguintes relações:

$$\begin{aligned} x_1^2 = x_2^2 = x_3^2 = z_1^2 = z_2^2 = z_3^2 = e; \\ \left[x_i, z_j\right] = \left[z_i, z_j\right] = e, \ \ i, j = 1, 2, 3; \\ \left[x_i, x_j\right] = z_{i+j-2}, \ i \neq j \in \{1, 2, 3\}; \end{aligned} \tag{2.5}$$

De novo aplicamos o Teorema de Schreier [15, Teorma 15.1.1]. Os automorfismos requeridos $a \mapsto a^h$ do grupo N são a aplicação identidade e o conjunto fator deve ser definido como segue:

$$(\overline{x}_1^{k_1}\overline{x}_2^{k_2}\overline{x}_3^{k_3}, \overline{x}_1^{r_1}\overline{x}_2^{r_2}\overline{x}_3^{r_3}) = z_1^{r_1k_2}z_2^{r_1k_3}z_3^{r_2k_3}, \quad \text{para todos } k_i, r_j \in F_2, \quad i, j = 1, 2, 3. \tag{2.6}$$

As condições do Teorema de Schreier são verificadas com cálculos diretos, portanto existe a extensão que procuramos. Resta provar que todo subgrupo abeliano A de G tem no máximo 2^4 elementos. Podemos supor $Z(G) \subseteq A$. Sejam dois elementos

$a, b \in A$ da forma $a = x_1^{k_1} x_2^{k_2} x_3^{k_3} z$ e $b = x_1^{r_1} x_2^{r_2} x_3^{r_3} z'$ tais que $z, z' \in Z(G), k_i, r_i \in F_2$, $i = 1, 2, 3$. Como estamos supondo A abeliano, $ab = ba$.

Como $\overline{a} = \overline{x}_1^{k_1} \overline{x}_2^{k_2} \overline{x}_3^{k_3}$ e $\overline{b} = \overline{x}_1^{r_1} \overline{x}_2^{r_2} \overline{x}_3^{r_3}$, então $\overline{x}_1^{k_1} \overline{x}_2^{k_2} \overline{x}_3^{k_3} \overline{x}_1^{r_1} \overline{x}_2^{r_2} \overline{x}_3^{r_3} = \overline{x}_1^{r_1} \overline{x}_2^{r_2} \overline{x}_3^{r_3} \overline{x}_1^{k_1} \overline{x}_2^{k_2} \overline{x}_3^{k_3}$. Pela equação (2.6), temos

$$z_1^{r_1 k_2} z_2^{r_1 k_3} z_3^{r_2 k_3} = z_1^{k_1 r_2} z_2^{k_1 r_3} z_3^{k_2 r_3} \quad \Longrightarrow \quad z_1^{r_1 k_2 - k_1 r_2} z_2^{r_1 k_3 - k_1 r_3} z_3^{r_2 k_3 - k_2 r_3} = e.$$

Daí obtemos o sistema

$$\begin{cases} r_1 k_2 - k_1 r_2 = 0 \\ r_1 k_3 - k_1 r_3 = 0 \\ r_2 k_3 - k_2 r_3 = 0 \end{cases}$$

e a seguinte matriz

$$\begin{pmatrix} k_1 & k_2 & k_3 \\ r_1 & r_2 & r_3 \end{pmatrix}.$$

Como todos os menores complementares da matriz são iguais a zero, a matriz sobre F_2 tem posto menor que 2. No corpo F_2, isso significa que ou qualquer uma das linhas do matriz é nula e então um dos elementos a, b pertence a $Z(G)$, ou ambas as linhas são iguais e daí $aZ(G) = bZ(G)$. Logo $|A| \leqslant 16$.

Como todo subgrupo abeliano tem no máximo $2^4 = 16$ elementos, tomamos A e B abelianos tais que $|A| = 2^4$ e $|B| = 2^4$, e ainda tais que $Z(G) \subset A$ e $Z(G) \subset B$. Como $|AB| \leqslant \dfrac{|A||B|}{|Z(G)|}$, então $|AB| \leqslant 2^5$. Como $|G| = 2^6$, então G não tem decomposição abeliana. □

Além do grupo que encontramos na demonstração acima, utilizando o GAP conseguimos verificar que existem outros 18 grupos de ordem 2^6 que não são decomponíveis. Para isso, utilizamos a função HasAbelianDecomposition apresentada anteriormente na Proposição 2.1.13 e exibimos todos os 19 grupos de ordem 2^6 que não tem decomposição abeliana.

```
gap> n:= Size(List(AllSmallGroups(Size,64), StructureDescription));
267

gap> for i in [1..n] do
    if not HasAbelianDecomposition(SmallGroup(64,i)) then
        Print (" Posicao: ", i, " - Nao tem decomposicao abeliana ",
                StructureDescription(SmallGroup(64,i)), "\n");
    fi;
 od;
 Posicao: 73 - Nao tem decomposicao abeliana (C2 x C2 x D8) : C2
 Posicao: 74 - Nao tem decomposicao abeliana ((C4 x C2) : C4) : C2
 Posicao: 75 - Nao tem decomposicao abeliana (C2 x ((C4 x C2) : C2)) : C2
 Posicao: 76 - Nao tem decomposicao abeliana (C4 x C2) : Q8
 Posicao: 77 - Nao tem decomposicao abeliana (C2 x (C4 : C4)) : C2
 Posicao: 78 - Nao tem decomposicao abeliana (C2 x (C4 : C4)) : C2
 Posicao: 79 - Nao tem decomposicao abeliana (C2 x C2 x C2) . (C2 x C2 x C2)
 Posicao: 80 - Nao tem decomposicao abeliana ((C4 x C2) : C4) : C2
 Posicao: 81 - Nao tem decomposicao abeliana (C2 x C2 x C2) . (C2 x C2 x C2)
 Posicao: 82 - Nao tem decomposicao abeliana (C2 x C2 x C2) . (C2 x C2 x C2)
 Posicao: 149 - Nao tem decomposicao abeliana ((C8 x C2) : C2) : C2
 Posicao: 150 - Nao tem decomposicao abeliana ((C4 x C2 x C2) : C2) : C2
 Posicao: 151 - Nao tem decomposicao abeliana (Q8 : C4) : C2
 Posicao: 170 - Nao tem decomposicao abeliana (C8 : C4) : C2
 Posicao: 171 - Nao tem decomposicao abeliana ((C8 x C2) : C2) : C2
 Posicao: 172 - Nao tem decomposicao abeliana (C2 x C2) . (C2 x D8) =
 (C4 x C2) . (C2 x C2 x C2)
 Posicao: 177 - Nao tem decomposicao abeliana ((C4 x C4) : C2) : C2
 Posicao: 178 - Nao tem decomposicao abeliana (C4 : Q8) : C2
 Posicao: 182 - Nao tem decomposicao abeliana C8 : Q8
```

2.2 Extensão de corpos

Nessa seção estabelecemos uma relação entre os G-códigos abelianos sobre um corpo E e sobre um subcorpo F de E. A referência principal é a Seção 2.2 da tese [17].

Lema 2.2.1. *[17, Lema 2.17] Se F é um subcorpo de um corpo E e I um ideal do anel de grupo FG, então EI é um ideal em EG e $PAut(I) = PAut(EI)$.*

Demonstração. Seja $B = \{e_1 = 1, ..., e_k\}$ uma base de E sobre F e $|G| = n$. Para um ideal I de FG, $EI = \sum_{i=1}^{k} e_i I$. De fato, podemos perceber que EI é um ideal de EG e $EI \cap FG = I$, comprovando a primeira parte do Lema.

Seja $\sigma \in PAut(I)$. Então σ é estendido a uma aplicação E-linear em EG. Se $x \in EI$ então $x = \sum_{i=1}^{k} e_i x_i$ para algum $x_i \in I, i \in \{1, ..., k\}$. Aplicando σ a x, obtemos

$$\sigma(x) = \sum_{i=1}^{k} e_i \sigma(x_i) \in EI,$$

já que $\sigma(x_i) \in I$, $i \in \{1, ..., k\}$. Isso prova que $PAut(I) \subseteq PAut(EI)$.

Agora considere $\tau \in PAut(EI)$. Novamente, consideramos τ como aplicação E-linear em EG. Uma vez que τ atua sobre G, $\tau(FG) = FG$, então

$$\tau(I) = \tau(EI \cap FG) = \tau(EI) \cap \tau(FG) = EI \cap FG = I.$$

Isto implica $PAut(EI) \subseteq PAut(I)$. Portanto, $PAut(I) = PAut(EI)$. □

Teorema 2.2.2. *[17, Teorema 2.18] Seja F um subcorpo de um corpo E e G um grupo. Se todo G-código sobre E é abeliano, então todo G-código sobre F também o é.*

Demonstração. Por hipótese, todo G-código sobre E é abeliano, assim, pelo Lema 2.1.2, existe um subgrupo abeliano transitivo que está contido em $PAut(EI)$. Além disso, pelo Lema 2.2.1, para qualquer ideal I de FG, o grupo $PAut(I)$ coincide com o grupo $PAut(EI)$. Logo, $PAut(I)$ também contém um subgrupo transitivo abeliano e novamente, pelo Lema 2.1.2, I é um código abeliano. □

O próximo teorema nos fornece um caso em que vale a recíproca do Teorema 2.2.2.

Teorema 2.2.3. *[17, Teorema 2.19] Sejam F um subcorpo de um corpo E e G um grupo. Suponhamos char $F \nmid |G|$ e*

$$FG \cong \bigoplus_{i=1}^{k} M_{d_i}(F) \tag{2.7}$$

Se todo G-código sobre F é abeliano, então todo G-código sobre E também o é.

Demonstração. De acordo com as hipóteses do Teorema 2.2.3, temos

$$\begin{aligned} EG &= EG \otimes_F F = E \otimes_F FG \\ &= E \otimes_F \left(\bigoplus_{i=1}^{k} M_{d_i}(F) \right) = \bigoplus_{i=1}^{k} E \otimes_F M_{d_i}(F) \\ &= \bigoplus_{i=1}^{k} M_{d_i}(E \otimes_F F) = \bigoplus_{i=1}^{k} M_{d_i}(E) \end{aligned}$$

Como todo anel de matrizes sobre um corpo é simples, então todo ideal J de EG é da forma $J = \bigoplus_{i \in S} M_{d_i}(E)$, com S um subconjunto de $\{1, ..., k\}$. Logo $J = EI$, com $I = \bigoplus_{i \in S} M_{d_i}(F)$. Pelo Lema 2.2.1, $PAut(I) = PAut(J)$ e, por hipótese, todo G-código sobre F é abeliano, o que significa que $PAut(I)$ contém um subgrupo transitivo abeliano pelo Lema 2.1.2, daí $PAut(J)$ também contém esse subgrupo transitivo abeliano. Portanto, o G-código sobre E definido pelo ideal J é abeliano. □

Observação: A condição (2.7) é satisfeita por uma família infinita de corpos F.

A Proposição 2.2.4 estabelece uma condição que nos fornece álgebras de grupo que atendam às hipóteses do Teorema 2.2.3.

Proposição 2.2.4. *[17, Proposição 2.20] Para cada primo p tal que $p \nmid |G|$, existe um número m tal que um corpo F, com $|F| = p^l$, satisfaz $FG \cong \bigoplus_{i=1}^{k} M_{d_i}(F)$ se, e somente se, $m|l$.*

Demonstração. Seja $F_0 = Z_p$, com $p \nmid |G|$. Pelo Teorema de Wedderburn-Artin para álgebras de grupo 1.1.12, $F_0G \cong \bigoplus_{i=1}^{r} M_{d_i}(F_i)$, com $F_1, ..., F_r$ extensões do corpo F_0. Como, para todo corpo F, de característica p tem-se

$$\begin{aligned} FG &= FG \otimes_{F_0} F_0 = F \otimes_{F_0} F_0 G \\ &= F \otimes_{F_0} \bigoplus_{i=1}^{r} M_{d_i}(F_i) = \bigoplus_{i=1}^{r} F \otimes_{F_0} M_{d_i}(F_i) = \bigoplus_{i=1}^{r} M_{d_i}(F \otimes_{F_0} F_i). \end{aligned}$$

Segue que $FG \cong \bigoplus_{i=1}^{k} M_{d_i}(F)$ é equivalente à seguinte condição.

$$\text{Para todo } i \in \{1, ..., r\},\ F \otimes_{F_0} F_i \cong F \oplus F ... \oplus F,$$

com o número de somandos isomorfos a F igual a $dim_{F_0}(F_i)$. O último isomorfismo se tem se, e somente se, o polinômio mínimo de um elemento primitivo de F_i sobre F_0 se decompõe sobre F [9, Capítulo 5]. Seja E o corpo de decomposição do produto dos polinômios mínimos dos elementos primitivos de $F_1, ..., F_r$ sobre F_0, e $m = dim_{F_0} E$, isto é, $|E| = p^m$. Então $FG \cong \bigoplus_{i=1}^{k} M_{d_i}(F)$ é equivalente ao isomorfismo de E com algum subcorpo E' de F, e esse isomorfismo existe se, e somente se, $m|l$. □

A Proposição 2.2.5 apresenta um contraexemplo para a recíproca do Teorema 2.2.2. A prova será omitida por não se tratar do foco deste trabalho, que são os códigos gerados pelos ideais de álgebras de grupo semissimples, mas pode ser verificada em [17].

Proposição 2.2.5. *[17, Proposição 2.21] Seja F o corpo F_2, $E = F_4$ sua extensão e G seja o grupo de quatérnio Q_8. Então todos os G-códigos à esquerda sobre F são abelianos, mas existem G-códigos sobre E que não são.*

Com os resultados apresentados nessa seção concluímos que se um G-código é abeliano sobre o corpo E, então ele também é abeliano para um subcorpo F de E. Entretanto, se um G-código é abeliano para um corpo F, não necessariamente será abeliano para uma extensão E do corpo F.

2.3 Idempotentes gerados por subgrupos

Esta seção apresenta a teoria que o GAP utiliza para calcular os idempotentes primitivos centrais. As referências principais são a documentação do GAP [3] e o artigo [15].

Seja G um grupo e F um corpo tal que $char\, F$ não divide a ordem de G. Se H é um subgrupo de G, então o elemento

$$\hat{H} = |H|^{-1} \sum_{x \in H} x$$

é um idempotente de FG se, e somente se, H é normal em G.

Se H é um subgrupo normal próprio de um subgrupo K de G então

$$\mathcal{E}(K, H) = \Pi_L(\hat{H} - \hat{L})$$

com L percorrendo os subgrupos normais de K que são minimais dentre os subgrupos normais de K contendo H. Por convenção, $\mathcal{E}(K, K) = \hat{K}$. O elemento $\mathcal{E}(K, H)$ é um idempotente de FG.

Se H e K são subgrupos de G, de modo que H é normal em K, então $e(G, K, H)$ denota a soma de todos os diferentes G-conjugados de $\mathcal{E}(K, H)$. O elemento $e(G, K, H)$ é central em FG. Em geral não é um idempotente, mas se os diferentes conjugados de $\mathcal{E}(K, H)$ são ortogonais, então $e(G, K, H)$ é um idempotente central de FG.

Um **idempotente central** e de um anel R é um idempotente que está no centro de R. Um **idempotente primitivo central** de um anel R é um idempotente central diferente de zero e que não pode ser escrito como a soma de dois idempotentes centrais não nulos de Re, ou equivalentemente, tal que Re não pode ser decomposto como um produto direto de dois ideais bilaterais não triviais.

A seguir, definimos pares de Shoda e pares de Shoda fortes, que serão usados para calcular os idempotentes.

Definição 2.3.1. *[15, Teorema 1.3] Seja G um grupo finito e um par (K, H) de subgrupos de G. Então (K, H) é um* ***par de Shoda*** *se, e somente se, as seguintes condições forem satisfeitas:*

1. $H \trianglelefteq K$;
2. K/H é cíclico;
3. Se $g \in G$ e $[K, g] \cap K \subseteq H$, então $g \in K$.

Corolário 2.3.2. *[15, Corolário 2.2] Se (H, K) é um par de Shoda de G, então existe um $a \in \mathbb{Q}$, necessariamente único, tal que $ae(G, K, H)$ é um* ***idempotente primitivo central*** *de $\mathbb{Q}G$.*

Os caracteres de grupos são utilizados para encontrar os idempotentes primitivos centrais, assim, temos a seguinte definição.

Definição 2.3.3. *O* ***caracter*** *de uma representação σ de um grupo G é a função χ^σ de G para o corpo de representação F dada por*

$$\chi^\sigma(g) = tr\,\sigma(g), \quad \text{para todo } g \in G.$$

Um grupo finito G é **monomial** se todos os caracteres complexos irredutíveis de G são monomiais, isto é, induzidos por um caracter linear de um subgrupo de G, conforme [4, §43].

Corolário 2.3.4. *[15, Corolário 2.3] Um grupo finito G é monomial se, e somente, se todos os idempotentes centrais primitivos de $\mathbb{Q}G$ tem a forma $ae(G, K, H)$, para $a \in \mathbb{Q}$ e (H, K) um par Shoda de G.*

Definição 2.3.5. *[15, Definição 3.1] Um* ***par de Shoda forte*** *de G é um par (K, H) de subgrupos de G que satisfazem as seguintes condições:*

1. $H \leq K \trianglelefteq N_G(H)$;
2. K/H é cíclico e um subgrupo abeliano maximal de $N_G(H)/H$;
3. para cada $g \in G \backslash N_G(H)$, $\mathcal{E}(K, H)\mathcal{E}(K, H)^g = 0$.

As seguintes afirmações são resultados provados em [15]. Seja (K, H) um par de Shoda forte de G, então (K, H) é um par de Shoda de G e $e(G, K, H)$ é um **idempotente primitivo central** de $\mathbb{Q}G$.

Seja G um grupo finito e χ um caracter irredutível de G. Diz-se que χ é **fortemente monomial** se houver um par de Shoda forte (K, H) de G e um caracter linear θ de K de G com núcleo H tal que $\chi = \theta^G$. O grupo G é **fortemente monomial** se todos os caracteres irredutíveis de G forem fortemente monomiais.

2.3.1 Classes ciclotômicas e pares de Shoda fortes

Seja G um grupo finito e F um corpo finito de ordem q, coprimo com a ordem de G.

Dado um número inteiro positivo n, coprimo com q, as **classes q-ciclotômicas módulo n** são o conjunto de classes de resíduos módulo n da forma

$$\{i, iq, iq^2, iq^3, ...\}.$$

As classes q-ciclotômicas módulo n formam uma partição do conjunto de classes de resíduos módulo n.

Uma **classe q-ciclotômica módulo n geradora** é uma classe ciclotômica que contém um gerador do grupo aditivo de classes de resíduos módulo n.

Sejam (K, H) um par de Shoda forte de G e $n = [K : H]$. Fixe uma raiz n-ésima primitiva da unidade ζ em alguma extensão de F e um elemento g de K tal que gH é um gerador de K/H. Seja C uma classe q-ciclotômica módulo n geradora. Defina

$$\mathcal{E}_C(K, H) = [K : H]^{-1} \hat{H} \sum_{i=0}^{n-1} tr(\zeta^{-ci}) g^i,$$

com c um elemento arbitrário de C e tr a função traço da extensão de corpos $F(\zeta)/F$. Então $\mathcal{E}_C(K, H)$ não depende da escolha de $c \in C$ e é um idempotente primitivo central

de FK.

Finalmente, seja $e_C(G,K,H)$ a soma dos diferentes G-conjugados de $\mathcal{E}_C(K,H)$. Então $eC(G,K,H)$ é um idempotente primitivo central de FG. Dizemos que $eC(G,K,H)$ é o **idempotente central primitivo realizado pelo par de Shoda forte** (K,H) **do grupo** G **e da classe** q**-ciclotômica** C.

Se G for fortemente monomial, então todo idempotente primitivo central de FG é realizável por algum par de Shoda forte de G e alguma classe ciclotômica C.

2.3.2 Aplicação dos Pares de Shoda no GAP

A função **PrimitiveCentralIdempotentsByStrongSP (FG)**, do pacote **wedderga** do GAP, deve ter como entrada FG, uma álgebra de grupo semissimples de um grupo finito G sobre um corpo finito F ou sobre o corpo $\mathbb{Q}$ dos racionais.

Se $F = \mathbb{Q}$, então a saída é a lista de idempotentes centrais primitivos da álgebra de grupo FG obtidos por pares de Shoda fortes de G. Se F é um corpo finito, a saída é a lista de idempotentes centrais primitivos de FG obtidos por pares de Shoda fortes (K,H) de G e das classes q-ciclotômicas módulo o índice de H em K.

A função **PrimitiveCentralIdempotentsByStrongSP(FG)** faz os cálculos dos idempotentes de acordo com a teoria aqui apresentada. Logo, se o grupo G não for fortemente monomial, então a lista resultante da função não conterá todos os idempotentes centrais primitivos dessa álgebra de grupo. Assim, o GAP enviará um aviso ao usuário na saída dessa função. Vide o exemplo a seguir da álgebra de grupo do grupo $SL(2,3)$ sobre o corpo finito de ordem 5, F_5.

Código 2.2: Exemplo onde a função PrimitiveCentralIdempotentsByStrongSP não calcula todos os idempotentes primitivos centrais existentes

```
1 FG := GroupRing( GF(5), SmallGroup(24,3) );
2 <algebra-with-one over GF(5), with 4 generators>
3 PrimitiveCentralIdempotentsByStrongSP( FG );
4
```

```
Wedderga: Warning!!!
The output is a NON-COMPLETE list of prim. central idemp.s of the
input!
[ (Z(5)^2)*<identity> of ...+(Z(5)^2)*f1+(Z(5)^2)*f2+(Z(5)^2)*f3+(Z(5)^2)*f4+(
    Z(5)^2)*f1^2+(Z(5)^2)*f1*f2+(Z(5)^2)*f1*f3+(Z(5)^2)*f1*f4+(Z(5)^2)*f2*f3+(
    Z(5)^2)*f2*f4+(Z(5)^2)*f3*f4+(Z(5)^2)*f1^2*f2+(Z(5)^2)*f1^2*f3+(Z(5)^
    2)*f1^2*f4+(Z(5)^2)*f1*f2*f3+(Z(5)^2)*f1*f2*f4+(Z(5)^2)*f1*f3*f4+(Z(5)^
    2)*f2*f3*f4+(Z(5)^2)*f1^2*f2*f3+(Z(5)^2)*f1^2*f2*f4+(Z(5)^2)*f1^2*f3*f4+(
    Z(5)^2)*f1*f2*f3*f4+(Z(5)^2)*f1^2*f2*f3*f4,
  (Z(5)^3)*<identity> of ...+(Z(5)^0)*f1+(Z(5)^3)*f2+(Z(5)^3)*f3+(Z(5)^3)*f4+(
    Z(5)^0)*f1^2+(Z(5)^0)*f1*f2+(Z(5)^0)*f1*f3+(Z(5)^0)*f1*f4+(Z(5)^3)*f2*f3+(
    Z(5)^3)*f2*f4+(Z(5)^3)*f3*f4+(Z(5)^0)*f1^2*f2+(Z(5)^0)*f1^2*f3+(Z(5)^
    0)*f1^2*f4+(Z(5)^0)*f1*f2*f3+(Z(5)^0)*f1*f2*f4+(Z(5)^0)*f1*f3*f4+(Z(5)^
    3)*f2*f3*f4+(Z(5)^0)*f1^2*f2*f3+(Z(5)^0)*f1^2*f2*f4+(Z(5)^0)*f1^2*f3*f4+(
    Z(5)^0)*f1*f2*f3*f4+(Z(5)^0)*f1^2*f2*f3*f4,
  (Z(5)^0)*<identity> of ...+(Z(5)^3)*f2+(Z(5)^3)*f3+(Z(5)^0)*f4+(Z(5)^
    3)*f2*f3+(Z(5)^3)*f2*f4+(Z(5)^3)*f3*f4+(Z(5)^3)*f2*f3*f4 ]
```

Capítulo 3

Os códigos em F_5S_4

No Capítulo 2 verificamos que a ordem do menor grupo que não é decomponível é 24 e encontramos que o S_4 e o $SL(2,3)$ não possuem decomposição abeliana. Neste capítulo buscaremos códigos de grupo não abelianos em F_5S_4, conforme o processo realizado no Capítulo 3 de [17]. Utilizamos o GAP para auxiliar as contas. As rotinas realizadas no GAP serão apresentadas ao longo da seção.

3.1 Análise dos códigos em F_5S_4

Seja $F = F_5$ um corpo finito de ordem 5 e $G = S_4$ o grupo simétrico das permutações sobre $\{1, 2, 3, 4\}$. Pela Teoria de Representações de Grupos, o anel de grupo $R := FG$ contém cinco ideais bilaterais minimais gerados pelos idempotentes primitivos centrais, como apresentado na rotina do GAP a seguir.

Código 3.1: Cálculos sobre a álgebra de grupo F_5S_4

```
1 LoadPackage("wedderga");
2
3 gap> S4:= SymmetricGroup(4);
4 Sym( [ 1 .. 4 ] )
5 gap> F5:=GF(5);
6 GF(5)
```

```
gap> R:= GroupRing(F5,S4);
<algebra-with-one over GF(5), with 2 generators>

gap> idemp:=PrimitiveCentralIdempotentsByStrongSP(R);
$[ (Z(5)^2)*()+(Z(5)^2)*(3,4)+(Z(5)^2)*(2,3)+(Z(5)^2)*(2,3,4)+(Z(5)^2)*
    (2,4,3)+(Z(5)^2)*(2,4)+(Z(5)^2)*(1,2)+(Z(5)^2)*(1,2)(3,4)+(Z(5)^2)*
    (1,2,3)+(Z(5)^2)*(1,2,3,4)+(Z(5)^2)*(1,2,4,3)+(Z(5)^2)*(1,2,4)+(Z(5)^2)*
    (1,3,2)+(Z(5)^2)*(1,3,4,2)+(Z(5)^2)*(1,3)+(Z(5)^2)*(1,3,4)+(Z(5)^2)*(1,3)
    (2,4)+(Z(5)^2)*(1,3,2,4)+(Z(5)^2)*(1,4,3,2)+(Z(5)^2)*(1,4,2)+(Z(5)^2)*
    (1,4,3)+(Z(5)^2)*(1,4)+(Z(5)^2)*(1,4,2,3)+(Z(5)^2)*(1,4)(2,3),(Z(5)^2)*()+
    (Z(5)^0)*(3,4)+(Z(5)^0)*(2,3)+(Z(5)^2)*(2,3,4)+(Z(5)^2)*
    (2,4,3)+(Z(5)^0)*(2,4)+(Z(5)^0)*(1,2)+(Z(5)^2)*(1,2)(3,4)+(Z(5)^2)*
    (1,2,3)+(Z(5)^0)*(1,2,3,4)+(Z(5)^0)*(1,2,4,3)+(Z(5)^2)*(1,2,4)+(Z(5)^2)*
    (1,3,2)+(Z(5)^0)*(1,3,4,2)+(Z(5)^0)*(1,3)+(Z(5)^2)*(1,3,4)+(Z(5)^2)*(1,3)
    (2,4)+(Z(5)^0)*(1,3,2,4)+(Z(5)^0)*(1,4,3,2)+(Z(5)^2)*(1,4,2)+(Z(5)^2)*
    (1,4,3)+(Z(5)^0)*(1,4)+(Z(5)^0)*(1,4,2,3)+(Z(5)^2)*(1,4)(2,3),
    (Z(5)^0)*()+(Z(5))*(2,3,4)+(Z(5))*(2,4,3)+(Z(5)^0)*(1,2)(3,4)+(Z(5))*
    (1,2,3)+(Z(5))*(1,2,4)+(Z(5))*(1,3,2)+(Z(5))*(1,3,4)+(Z(5)^0)*(1,3)(2,4)+
    (Z(5))*(1,4,2)+(Z(5))*(1,4,3)+(Z(5)^0)*(1,4)(2,3),(Z(5)^0)*()+(Z(5))*(3,4)+
    (Z(5))*(2,3)+(Z(5))*(2,4)+(Z(5))*(1,2)+(Z(5)^3)*(1,2)(3,4)+(Z(5)^3)*(1,2,3,4)+
    (Z(5)^3)*(1,2,4,3)+(Z(5)^3)*(1,3,4,2)+(Z(5))*(1,3)+(Z(5)^3)*(1,3)(2,4)+
    (Z(5)^3)*(1,3,2,4)+(Z(5)^3)*(1,4,3,2)+(Z(5))*(1,4)+(Z(5)^3)*(1,4,2,3)+
    (Z(5)^3)*(1,4)(2,3),(Z(5)^0)*()+(Z(5)^3)*(3,4)+(Z(5)^3)*(2,3)+(Z(5)^3)*(2,4)
    +(Z(5)^3)*(1,2)+(Z(5)^3)*(1,2)(3,4)+(Z(5))*(1,2,3,4)+(Z(5))*(1,2,4,3)+
    (Z(5))*(1,3,4,2)+(Z(5)^3)*(1,3)+(Z(5)^3)*(1,3)(2,4)+(Z(5))*(1,3,2,4)+
    (Z(5))*(1,4,3,2)+(Z(5)^3)*(1,4)+(Z(5))*(1,4,2,3)+(Z(5)^3)*(1,4)(2,3) ]$
```

Reescrevendo os idempotentes centrais primitivos considerando:

$$0 * Z(5) = 0,\ \ Z(5)\hat{}0 = 1 = -4,\ \ Z(5) = 2 = -3$$
$$Z(5)\hat{}2 = 2^2 = 4 = -1,\ \ Z(5)\hat{}3 = 2^3 = 8 = 3 = -2,$$

temos

$$
\begin{aligned}
e_0 &= -() - (3,4) - (2,3) - (2,3,4) - (2,4,3) - (2,4) - (1,2) - (1,2)(3,4) - (1,2,3) \\
&- (1,2,3,4) - (1,2,4,3) - (1,2,4) - (1,3,2) - (1,3,4,2) - (1,3) - (1,3,4) - (1,3)(2,4) \\
&- (1,3,2,4) - (1,4,3,2) - (1,4,2) - (1,4,3) - (1,4) - (1,4,2,3) - (1,4)(2,3);
\end{aligned}
$$

$$
\begin{aligned}
e_4 &= -() + (3,4) + (2,3) - (2,3,4) - (2,4,3) + (2,4) + (1,2) - (1,2)(3,4) - (1,2,3) \\
&+ (1,2,3,4) + (1,2,4,3) - (1,2,4) - (1,3,2) + (1,3,4,2) + (1,3) - (1,3,4) - (1,3)(2,4) \\
&+ (1,3,2,4) + (1,4,3,2) - (1,4,2) - (1,4,3) + (1,4) + (1,4,2,3) - (1,4)(2,3);
\end{aligned}
$$

$$
\begin{aligned}
e_2 &= () - 3(2,3,4) - 3(2,4,3) + (1,2)(3,4) - 3(1,2,3) - 3(1,2,4) - 3(1,3,2) - 3(1,3,4) \\
&+ (1,3)(2,4) - 3(1,4,2) - 3(1,4,3) + (1,4)(2,3);
\end{aligned}
$$

$$
\begin{aligned}
e_1 &= 3() + (3,4) + (2,3) + (2,4) + (1,2) - (1,2)(3,4) - (1,2,3,4) - (1,2,4,3) - (1,3,4,2) \\
&+ (1,3) - (1,3)(2,4) - (1,3,2,4) - (1,4,3,2) + (1,4) - (1,4,2,3) - (1,4)(2,3);
\end{aligned}
$$

$$
\begin{aligned}
e_3 &= 3() - (3,4) - (2,3) - (2,4) - (1,2) - (1,2)(3,4) + (1,2,3,4) + (1,2,4,3) + (1,3,4,2) \\
&- (1,3) - (1,3)(2,4) + (1,3,2,4) + (1,4,3,2) - (1,4) + (1,4,2,3) - (1,4)(2,3).
\end{aligned}
$$

Nos idempotentes e_1 e e_3 o resultado obtido foi multiplicado por 3. Essa operação é permitida, pois não altera o idempotente.

Assim, os ideais bilaterais minimais e suas respectivas dimensões são as seguintes:

Código 3.2: Cálculo dos ideais e suas dimensões

```
gap> ideale0:=R*idemp[1];
<left ideal in <algebra-with-one over GF(5), with 2 generators>, (1 generators)>
gap> ideale4:=R*idemp[2];
<left ideal in <algebra-with-one over GF(5), with 2 generators>, (1 generators)>
gap> ideale2:=R*idemp[3];
<left ideal in <algebra-with-one over GF(5), with 2 generators>, (1 generators)>
gap> ideale1:=R*idemp[4];
<left ideal in <algebra-with-one over GF(5), with 2 generators>, (1 generators)>
gap> ideale3:=R*idemp[5];
<left ideal in <algebra-with-one over GF(5), with 2 generators>, (1 generators)>

gap> Dimension(ideale0);
1
gap> Dimension(ideale4);
1
gap> Dimension(ideale2);
4
gap> Dimension(ideale1);
9
gap> Dimension(ideale3);
9
```

Também podemos verificar a dimensão desses ideais da seguinte forma:

Código 3.3: Outra maneira de calcular a dimensões dos ideais

```
gap> D:=DirectSumDecomposition(R);;
gap> List(D,Dimension);
[ 9, 9, 4, 1, 1 ]
```

Isso porque a álgebra de grupo pode ser escrita como a soma direta dos ideais minimais (Teorema de Wedderburn Artin).

Teorema 3.1.1. *[17, Teorema 3.1] Os códigos $C_1 = Re_1$ e $C_3 = Re_3$ são não abelianos. Os códigos $C_0 = Re_0$, $C_2 = Re_2$ e $C_4 = Re_4$ são abelianos.*

Demonstração. Perceba que qualquer ciclo $\sigma \in S_G = S_{S_4}$ de ordem 24 satisfaz $\sigma(e_0) = e_0$. Se tormarmos um ciclo $\sigma \in S_G$ que intercambie elementos pares e ímpares de G, então $\sigma(e_4) = -e_4 \in Re_4$, e $\langle\sigma\rangle$ é um subgrupo transitivo abeliano de S_G contido em $PAut(Re_0)$ e em $PAut(Re_4)$. Pelo Lema 2.1.2, os códigos C_0 e C_4 são abelianos.

Resta provar que o código $C_2 = Re_2$ é abeliano. Para isso, considere o subgrupo de Klein $K = V_4 \subset S_4 = G$ e apresentemos G como a união de classes módulo K:

$$G = Ka_0 \cup Ka_1 \cup ... \cup Ka_5.$$

Três das permutações $a_0, a_1, a_2, a_3, a_4, a_5$ são pares e três são ímpares. Assim, assumimos $(-1)^{|a_i|} = (-1)^i$, e $a_0 = e$ para $i \in \{0, ..., 5\}$. Seja $\sigma_0 \in S_K$ qualquer ciclo de ordem 4 e $\sigma \in S_G$ definido por $\sigma(xa_i) = \sigma_0(x)a_i$, para todo $x \in K$ e $i \in \{0, ..., 5\}$. É claro que $o(\sigma) = 4$ e $\sigma|K = \sigma_0$. Consideremos também uma permutação $\tau \in S_G$ definida por $\tau(xa_i) = (x)a_{i+1 \, mod \, 6}$, para quaisquer $x \in K$ e $i \in \{0, ..., 5\}$. Agora temos $o(\tau) = 6$. Portanto, $\sigma\tau = \tau\sigma$ e $\langle\sigma, \tau\rangle$ é um grupo transitivo abeliano em S_G. Podemos comprovar de forma direta (ou com a ajuda de um computador), que $\langle\sigma, \tau\rangle \subseteq PAut(Re_2)$, pelo que Re_2 também definide um código abeliano.

Para provar que Re_1 e Re_3 definem códigos não abelianos, precisaremos do auxílio da ferramenta computacional GAP.

Primeiro calculamos a distribuição de peso para os dois códigos (que será a mesma para ambos os ideais) e obtemos a Tabela 3.1.

Em seguida, testamos todos os códigos abelianos de comprimento 24 sobre F_5, o corpo finito com 5 elementos. Para unificar o algoritmo da pesquisa, notamos que qualquer grupo abeliano A de ordem 24 é produto direto de grupos cíclicos $\langle a\rangle_m \times \langle b_1\rangle_2 \times ... \times \langle b_k\rangle_2$, com $0 \leqslant k \leqslant 2$ e $m = 24/2^k \in \{24, 12, 6\}$ (m é o expoente do grupo abeliano A).

Seja $B = F[\langle b_1\rangle_2 \times ... \times \langle b_k\rangle_2]$ (se $k = 0$, tomemos $B = F$). Assim, o anel de grupo FA pode ser apresentado como $FA = B\langle a\rangle_m$. Podemos ver que todos os ideais minimais do anel B tem dimensão 1 e são gerados por elementos mutuamente ortogonais

Tabela 3.1: Distribuição de pesos das palavras do código gerado pelo ideal de dimensão 9

Peso d	Número de palavras de peso d	Peso d	Número de palavras de peso d
0	1	17	190080
8	324	18	320640
10	144	19	365184
12	5520	20	437952
13	2304	21	245760
14	23808	22	158400
15	23328	23	47232
16	111840	24	20608

$f_i = (e \pm b_1)(e \pm b_2)...(e \pm b_k), i = 1, ..., 2^k$. Seja $\langle b_1 \rangle_2 \times ... \times \langle b_k \rangle_2 = \{h_1, ..., h_K\}$, com $K = 2^k$, $f_j = \sum_{i=1}^{K} \varepsilon_{ij} h_i$, com $\varepsilon_{ij} = \pm 1 \in F$. Todo ideal $I \lhd B\langle a \rangle_m$ tem uma decomposição na forma $I = \sum_{j=1}^{K} I_j f_j$, com $I_j \lhd F\langle a \rangle_m$. Observe que, para palavras arbitrárias $w_j \in I_j$, a seguinte igualdade é válida:

$$\left|\left| \sum_{j=1}^{K} w_j f_j \right|\right| = \left|\left| \sum_{j=1}^{K} w_j \sum_{i=1}^{K} \varepsilon_{ij} h_i \right|\right| =$$

$$\left|\left| \sum_{i=1}^{K} \left(\sum_{j=1}^{K} \varepsilon_{ij} w_j \right) h_i \right|\right| = \sum_{i=1}^{K} \left|\left| \left(\sum_{j=1}^{K} \varepsilon_{ij} w_j \right) \right|\right|, \tag{3.1}$$

uma vez que os suportes de $w_j h_i$ e $w_s h_t$ têm interseção vazia para $t \neq i$. Aqui $||w||$ significa o número de elementos no suporte da palavra w.

Assim, para listar todos os ideais em $B\langle a \rangle_m$ de uma dada dimensão k, é suficiente listar os ideais de $F\langle a \rangle_m$. A descrição destes ideais, isto é, dos códigos cíclicos de comprimento m sobre o corpo F, é conhecida, uma vez que um código cíclico pode ser visto como um ideal no anel de polinômios módulo $\langle x^m - 1 \rangle$ (vide [11, Capítulos 7, 8]).

Seja $x^m - 1 = \varphi_1^{(m)}(x)...\varphi_{r_m}^{(m)}(x)$ a decomposição canônica do polinômio $x^m - 1$ no produto de polinômios irredutíveis unitários sobre F e

$$g_j^{(m)}(x) = \frac{x^m - 1}{\varphi_j^{(m)}(x)}.$$

$$g_j^{(m)}(a),\ a g_j^{(m)}(a),\ ...,\ a^{d_j^{(m)}} g_j^{(m)}(a), \tag{3.2}$$

com $d_j = m - deg\, g_j^{(m)}(x) - 1$. Para os valores de $m \in \{24, 12, 6\}$, temos as seguintes decomposições sobre F:

$$\begin{aligned} x^6 - 1 &= (x-1)(x+1)(x^2 - x + 1)(x^2 + x + 1); \\ x^{12} - 1 &= (x^6 - 1)(x+2)(x+3)(x^2 + 3x - 1)(x^2 + 2x - 1); \\ x^{24} - 1 &= (x^{12} - 1)(x^2 + 2)(x^2 + 3)(x^2 - x + 2) \\ &\quad \times (x^2 + x + 2)(x^2 + 2x + 3)(x^2 - 2x + 3). \end{aligned} \tag{3.3}$$

Algumas relações de simetria podem ser aplicadas para reduzir o número de ideais a serem verificados. Analogamente, não calculamos a distribuição completa de pesos para os códigos abelianos em questão, mas paramos a busca no momento em que, para algum peso w, o número de palavras de peso w já encontradas supera o número dessas palavras para o código $C_1 = Re_1$. □

Esse teorema também pode ser escrito como no Teorema 3.1.2 (Teorema 2.1 de [18]). Nele vemos a demonstração computacional completa de que os códigos C_1 e C_3 são não abelianos.

Teorema 3.1.2. *[18, Teorema 2.1] Os códigos correspondentes aos ideais minimais de dimensão 9 do anel R são não abelianos.*

Demonstração. Para essa demonstração utilizaremos a função em GAP criada e apresentada em [18] que calcula a distribuição de pesos de um ideal I.

Código 3.4: Função que calcula o peso das palavras do código

```
1 DistribuicaoDePeso:=function(I,R)
2 local wlist, k, j, d, x, V, B, mf;
```

```
mf:=Size(LeftActingDomain(R))-1;
wlist:=List([0..Dimension(R)],x->0);
wlist[1]:=1;
d:=Dimension(I);
B:=BasisVectors(Basis(I));
for j in [1..d] do
    V:=SubspaceNC(R,B{[(j+1)..d]});
    for x in V do
        k:=Size(CoefficientsAndMagmaElements(B[j]+x))/2+1; #calculando o numero de elementos
        # no coeficiente das palavras
        wlist[k]:=wlist[k]+mf;
od;
od;
return wlist;
end;
```

Daí calculamos a distribuição de pesos dos ideais gerados pelos idempotentes e_1 e e_4, ambos com dimensão 9. A seguir apresentamos o resultado obtido na rotina do GAP. O mesmo também pode ser verificado na Tabela 3.2.

```
gap> distribuicaoIdeale1:=DistribuicaoDePeso(ideale1, R);
[ 1, 0, 0, 0, 0, 0, 0, 0, 324, 0, 144, 0, 5520, 2304, 23808, 23328, 111840, 190080,
320640, 365184, 437952, 245760, 158400, 47232, 20608 ]
gap> distribuicaoIdeale3:=DistribuicaoDePeso(ideale4, R);
[ 1, 0, 0, 0, 0, 0, 0, 0, 324, 0, 144, 0, 5520, 2304, 23808, 23328, 111840, 190080,
320640, 365184, 437952, 245760, 158400, 47232, 20608 ]
```

Em [6] vemos que dois códigos que são equivalentes têm a mesma distribuição de pesos, mas a recíproca dessa afirmação não é verdade, isto é, dois códigos que têm a mesma distribuição de pesos não são necessariamente equivalentes.

No seguinte passo, faremos o processo realizado em [18] para testar todos os códigos abelianos de comprimento 24 em F_5 e verificar se algum código poderia ser equivalente a um dos códigos de dimensão 9 que estamos trabalhando. Com base nas seguintes observações, podemos reduzir os cálculos:

Tabela 3.2: Distribuição de pesos das palavras do código gerado pelo ideal de dimensão 9

Peso d	Número de palavras de peso d	Peso d	Número de palavras de peso d
0	1	17	190080
8	324	18	320640
10	144	19	365184
12	5520	20	437952
13	2304	21	245760
14	23808	22	158400
15	23328	23	47232
16	111840	24	20608

1. A ação de um automorfismo de grupo em qualquer grupo pode ser estendida ao anel de grupo e esta extensão é um automorfismo que preserva o peso do anel de grupo;

2. Se durante o cálculo da distribuição de peso de algum ideal acontecer que, para algum peso w, o número de palavras já encontradas com esse peso ultrapassa o número palavras de peso w da distribuição de peso já conhecida, na variável *distribuicaoIdeale*1, então a distribuição de pesos deste ideal é diferente da distribuição de pesos presente na variável *distribuicaoIdeale*1. Logo, podemos parar o cálculo para este ideal, pois isso significa que os códigos não podem ser equivalentes.

Para realizar esses cálculos usamos as seguintes funções no GAP apresentadas em [18].

1. A função *StandardIsomorphismsOfAGroupRing* que transforma automorfismos de um grupo em automorfismos do anel de grupo.

```
1 StandardIsomorphismsOfAGroupRing:=function(R,HH)
2 local H, h, f, x, y, B1, B2, C, n;
3 H:=[];
4 B1:=BasisVectors(Basis(R));
```

```
5  n:=Size(B1);
6  C:=List(B1, x->(CoefficientsAndMagmaElements(x)[1]));
7  for h in HH do
8      B2:=List([1..n], x->B1[Position(C,Image(h,C[x]))]);
9      #Print(B2, "\n");
10     Add(H, AlgebraHomomorphismByImagesNC( R, R, B1, B2 ));
11 od;
12 return H;
13 end;
```

2. A função *CombinationsOfGivenSum* que produz uma lista de ideais de determinada dimensão k. Cada um desses ideais é definido como uma soma de ideais minimais cujas dimensões são enumeradas na lista l.

```
1  CombinationsOfGivenSum:=function(l,k)
2  local AllCombList, n, s; n:=Size(l);
3  AllCombList:=Combinations([1..n]);
4  return Filtered(AllCombList, x->(Sum(List(x, i->l[i]))=k));
5  end;
```

3. A função *PermutationsOfComponents* que enumera as permutações do conjunto de ideais minimais induzidos pelo conjunto H de automorfismos do grupo (mais precisamente pelo conjunto HH de extensões dos automorfismos de grupo em automorfismos do anel de grupo).

```
1  PermutationsOfComponents:=function(R,H,DSD)
2  local x, y, h, HH, PL, l, B, I, II, pl;
3  PL:= [[]];
4  l:=Size(DSD);
5  PL:= [[1..l]]; #identity permutation must present
6  HH:=StandardIsomorphismsOfAGroupRing(R,H);
7  for h in HH do
8      pl:=[];
9      for I in DSD do
10         B:=BasisVectors(Basis(I));
11         II:=Ideal(R,List(B,y->Image(h,y)));
12         Add(pl,Position(DSD,II));
```

```
13      od;
14      if not pl in PL then Add(PL,pl);
15      fi;
16  od;
17  return PL;
18  end;
```

4. A função *IsMinimalCombination* que verifica se alguma permutação de ideais minimais contidos na lista de permutação PL, da função *IsMinimalCombination*, transforma o conjunto de ideais minimais em algum conjunto que é menor do que o dado.

   Ao procurar um ideal com distribuição de peso idêntica, é suficiente considerar apenas aqueles que correspondam a conjuntos lexicograficamente mínimos de ideais minimais.

```
1  IsMinimalCombination:=function(L, PL)
2  local x, i;
3  for x in PL do
4      for i in L do
5          if x[i]>i then
6              break;
7          else if x[i]<i then
8                  return false;
9              fi;
10         fi;
11     od;
12 od;
13 return true;
14 end;
```

5. A função *EqualWeightDistribution* que compara a distribuição de peso de um ideal I com uma distribuição de peso WD dada.

```
1  EqualWeightDistribution:=function(I,R, WD)
2  local wlist, k, j, d, x, V, B, mf;
3  mf:=Size(LeftActingDomain(R))-1;
```

```
wlist:=List([0..Dimension(R)],x->0);
wlist[1]:=1;
d:=Dimension(I);
B:=BasisVectors(Basis(I));
for j in [1..d] do
    V:=SubspaceNC(R,B{[(j+1)..d]});
    for x in V do
        k:=Size(CoefficientsAndMagmaElements(B[j]+x))/2+1;
        wlist[k]:=wlist[k]+mf;
        if wlist[k]>WD[k] then return false;
        fi;
    od;
od;
return true;
end;
```

Agora verificaremos o teorema utilizando as funções apresentadas anteriormente. Note que WD1 = $distribuicaoIdeale1 = distribuicaoIdeale4$.

```
G:=SymmetricGroup(4);
F:=GF(5);
R:=GroupRing(F,G);
D:=DirectSumDecomposition(R);;
WD1:=DistribuicaoDePesos(D[1],R);
allab:=AllSmallGroups(Size,24,IsAbelian,true);;
dsdperm:=[];
dsd:=[];
for A in allab do
    R:=GroupRing(F,A);;
    dsd:=DirectSumDecomposition(R);;
    dsddim:=List(dsd,Dimension);;
    CL:=CombinationsOfGivenSum(dsddim,9);;
    H:=AutomorphismGroup(A);;
    dsdperm:=PermutationsOfComponents(R,H,dsd);;
    RCL:=Filtered(CL,x->IsMinimalCombination(x,dsdperm));;
    for C in RCL do I:=Sum(List(C, x->dsd[x]));;
        if EqualWeightDistribution(I,R, WD1) then
```

```
19          Print("Equal weight distribution found\n");; break;
20          fi;
21      od;
22  od;
```

O código realizado no GAP rodou durante 2 horas e 30 min e não exibiu resultado, logo não foi encontrado código com a distribuição de peso da Tabela 3.2. Assim, não existem códigos abelianos que possam ser equivalentes aos códigos gerados pelos ideais minimais de dimensão 9 do anel R. Portanto, os códigos correspondentes aos ideais minimais de dimensão 9 no anel R não são abelianos. □

Agora façamos uma pequena análise sobre os códigos encontrados. Utilizamos o GAP para calcular as demais distribuições de peso, dos códigos abelianos gerados por e_0, e_2 e e_4.

```
1
2  gap> distribuicaoIdeale0:=DistribuicaoDePeso(ideale0, R);
3  [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4 ]
4
5  gap> distribuicaoIdeale2:=DistribuicaoDePeso(ideale2, R);
6  [ 1, 0, 0, 0, 0, 0, 0, 0, 24, 0, 0, 0, 24, 0, 0, 0, 144, 0, 0, 0, 288, 0, 0, 0, 144 ]
7
8  gap> distribuicaoIdeale4:=DistribuicaoDePeso(ideale4, R);
9  [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4 ]
```

A Tabela 3.3 apresenta os resultados de todos os parâmetros dos códigos gerados por e_0, e_1, e_2, e_3 e e_4.

Note que apesar dos códigos abelianos gerados por e_0 e e_4 terem distância mínima maior que os demais, eles possuem dimensão 1, o que significa pouca variedade de palavras no código (os códigos gerados por e_1 e e_3 tem muito mais palavras, pois tem dimensão 9). Já o código abeliano gerado por e_2 possui dimensão maior que os outros abelianos, porém menor que as dimensões dos códigos não abelianos e distância mínima igual. Portanto, neste caso, os códigos não abelianos possuem parâmetros melhores que os abelianos.

Tabela 3.3: Tabela de distribuição de pesos dos códigos

gerador	(n, k, d)	$w_H(x)$
e_0	(24, 1, 24)	$1 + 4x^{24}$
e_1	(24, 9, 8)	$1 + 324x^8 + 144x^{10} + 5520x^{12} + 2304x^{13} + 23808x^{14} + 23328x^{15}$ $+111840x^{16} + 190080x^{17} + 320640x^{18} + 365184x^{19} + 437952x^{20}$ $+245760x^{21} + 158400x^{22} + 47232x^{23} + 20608x^{24}$
e_2	(24, 4, 8)	$1 + 24x^8 + 24x^{12} + 144x^{16} + 288x^{20} + 144x^{24}$
e_3	(24, 9, 8)	$1 + 324x^8 + 144x^{10} + 5520x^{12} + 2304x^{13} + 23808x^{14} + 23328x^{15}$ $+111840x^{16} + 190080x^{17} + 320640x^{18} + 365184x^{19} + 437952x^{20}$ $+245760x^{21} + 158400x^{22} + 47232x^{23} + 20608x^{24}$
e_4	(24, 1, 24)	$1 + 4x^24$

Capítulo 4

O processo de Decodificação

Como já mencionado, o grande problema da Teoria de Códigos é encontrar códigos com boas propriedades, isto é, códigos cujo comprimento não seja muito grande, pois comprimentos grandes requerem custos de transmissão mais altos; cuja dimensão seja grande o suficiente para possibilitar um grande número de palavras no código e cuja distância mínima seja grande, possibilitando a correção de muitos erros.

Outra grande questão na Teoria de Códigos é o processo de decodificação, pois o processo de correção de erros, isto é, a busca pela palavra do código mais próxima da palavra recebida pode demorar muito e assim ter um alto custo computacional.

Veremos aqui alguns resultados importantes para compreendermos os processos de decodificação. Este capítulo tem como referência principal o Capítulo 4 da tese [17].

4.1 Um método de decodificação

Inicialmente lembramos um aperfeiçoamento do método de decodificação inventado por D. Slepian, do Laboratório Bell na década de 60 do século XX.

Seja e o vetor erro, c o vetor transmitido e r o vetor recebido, temos

$$e = r - c.$$

Assim, o peso do vetor erro e corresponde ao número de erros cometidos no processo de transmissão e recepção.

Para o processo de decodificação também é utilizada a noção de síndrome. Como mencionado na Definição 1.2.17, a **síndrome de** r é o vetor $\mathcal{H}r^t$, com $\mathcal{H}$ a matriz teste de paridade (também chamada matriz de controle). De fato, $\mathcal{H}r^t = 0$ se, e somente se, r é um vetor do código. Desta maneira,

$$\mathcal{H}c^t = 0 \qquad \text{e} \qquad \mathcal{H}e^t = \mathcal{H}(r^t - c^t) = \mathcal{H}r^t - \mathcal{H}c^t = \mathcal{H}r^t.$$

Proposição 4.1.1. *[17, Proposição 4.1] A síndrome do vetor recebido r é uma combinação linear das colunas de $\mathcal{H}$ correspondentes às posições em que ocorreram erros.*

Considere um código com capacidade corretora de pelo menos um erro. Para uma palavra deste código na qual temos a ocorrência de somente um erro, seja $e = (0, ..., 0, e_i, 0, ..., 0)$ o vetor erro cuja síndrome deve ser um múltiplo da i-ésima coluna de $\mathcal{H}$, $\mathcal{H}e = h_i e_i = \mathcal{H}r$. Deste modo, a posição da coluna h_i é a posição do erro e podemos determinar facilmente a mensagem enviada, já que $c = r - e$.

4.2 Matriz Geradora de um código

Lembremos que a matriz geradora de um código C sobre F_q, com parâmetros (n, k, d) e base $\mathcal{B} = \{v_1, ..., v_k\}$, é a matriz cujas linhas são os vetores $v_i = (v_{i1}, ..., v_{in})$, $i = 1, ..., k$. Utilizando as operações definidas na Subseção 1.2.2, conseguimos encontrar uma matriz geradora de um código equivalente que esteja na forma padrão, isto é, que sejam da forma $\mathcal{G} = (Id_k|A)$.

Considere a álgebra de grupo de um grupo finito G sobre um corpo finito F. Pelo Teorema 1.1.12, um anel semissimples é a soma direta dos ideais à esquerda minimais desse anel. Além disso, segundo [14, Teorema 2.5.10] todo ideal minimal à esquerda L é gerado por um idempotente e, isto é, $L = FGe$. Assim, existem elementos idempotentes dois a dois ortogonais, de modo que $1 = e_1 + ... + e_n$ e $FG = FGe_1 \oplus ... \oplus FGe_n$, com FGe_i, $i = 1, ..., n$, ideais à esquerda minimais de FG.

Desta maneira, considerando um ideal à esquerda I gerado por um idempotente e, para encontrar os ideais à esquerda irredutíveis que são somandos diretos, basta fazer o produto ee_i para $i = 1, ..., n$. Assim, o ideal minimal FGe_i aparece na decomposição de FGe se, e somente se, $ee_i \neq 0$. Além disso, o anulador $I^{\perp}$ de I é a soma direta dos ideais minimais à esquerda que não aparecem na decomposição de I, ou seja, a soma direta dos ideais FGe_i, tais que $ee_i = 0$.

Desta maneira, temos a seguinte proposição.

Proposição 4.2.1. *[17, Proposição 4.2] Seja e um elemento idempotente que gera o código de grupo à esquerda I de FG. Então há k elementos $u_1, u_2, ..., u_k$ que são linearmente independentes em FG e tais que os elementos $u_1e, u_2e, ..., u_ke$ formam uma F-base de I.*

A questão agora é como calcular a dimensão de um código de grupo à esquerda, isto é, de um ideal I à esquerda de FG, sabendo que e é um gerador idempotente de I.

Este assunto está relacionado à Teoria de Representações de Grupos. Toda representação $D^{(n_o)} : G \to GL_{n_o}(L)$ de um grupo G induz uma representação $D^{(n_o)} : FG \to GL_{n_o}(L)$ da álgebra de grupo FG, aqui denotada da mesma maneira, dada por $D^{(n_o)}(a) = \sum_{i=1}^{|G|} f_i D^{n_o}(g_i)$, com $(a) = \sum_{(i=1)}^{n} f_i g_i$. Aqui L é uma extensão de F.

A representação $D^{(n_o)}$ de FG é irredutível ou redutível se a representação $D^{(n_o)}$ de G o for.

O Teorema 4.2.2 a seguir foi apresentado em [17] de acordo com o Teorema 1 de [5] e nos fornece uma maneira de calcular a dimensão de um ideal I.

Teorema 4.2.2. *Seja I um ideal à esquerda de FG, com geradores $f_1, ..., f_t$. Seja M a matriz cuja j-ésima linha é dada pelos coeficientes de $g_j f_i$ em relação à base $G = \{g_1, ..., g_n\}$ de FG. A matriz*

M assume a seguinte forma

$$M = \begin{bmatrix} M_1 \\ \vdots \\ M_t \end{bmatrix} = \begin{bmatrix} g_1f_1 & g_1f_2 & \cdots & g_1f_t \\ g_2f_1 & g_2f_2 & \cdots & g_2f_t \\ \vdots & \vdots & \cdots & \vdots \\ g_nf_1 & g_nf_2 & \cdots & g_nf_t \end{bmatrix}.$$

Então a dimensão de I como F-espaço vetorial é igual ao posto de M.

Demonstração. Basta notar que os elementos g_if_k formam um sistema F-gerador de I. □

Consideremos agora os seguintes resultados.

Teorema 4.2.3. *[4, Teorema 27.22], Seja G um grupo finito e K um corpo algebricamente fechado tal que char K ∤ |G|. Então o número de representações irredutíveis de G não isomorfas é igual ao número de classes de conjugação de G.*

Teorema 4.2.4. *[10, Teorema 2.1.12] Qualquer corpo é um corpo de decomposição para S_n.*

Assim, na álgebra de grupo semissimples FS_n, com F um corpo qualquer, o número de representações irredutíveis, consequentemente o número de ideais centrais minimais, é igual ao número de classes de conjugação do grupo S_n.

Seja o código $C = FGe$ correspondente ao ideal gerado pelo idempotente e. Considere $\{u_1e, ..., u_ke\}$ uma base do código C, logo $\{u_1, ..., u_k\}$ são linearmente independentes. A matriz geradora $\mathcal{G}$ é a matriz cuja j-ésima coluna corresponde às coordenadas $u_je = u_{1j}g_1 + u_{2j}g_2 + \cdots + u_{nj}g_n$.

$$\mathcal{G} = \begin{pmatrix} u_1e & u_2e & \ldots & u_ke \end{pmatrix} = \begin{pmatrix} u_{11} & u_{12} & \ldots & u_{1k} \\ u_{21} & u_{22} & \ldots & u_{2k} \\ \vdots & \vdots & \ldots & \vdots \\ u_{n1} & u_{n2} & \ldots & u_{nk} \end{pmatrix}$$

Assim, uma palavra $c_1g_1 + ... + c_ng_n \in FG$ é um elemento do código se, e somente se,

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}, \quad c = \mathcal{G}x, \, com \, x = (x_1, ..., x_k)^T.$$

Desta maneira, temos o conceito de matriz geradora na forma padrão, neste caso $\tilde{\mathcal{G}} = (I_k|A)^T$.

Note que, em [17], temos as matrizes transpostas das matrizes mostradas por [8] no Capítulo 1, mas vale ressaltar que os resultados são equivalentes. Optamos por utilizar os dois para evitar possíveis erros ao converter de uma notação para outra.

4.3 Ideal de controle e matriz de controle

Conforme [8], a matriz de controle ou matriz teste de paridade $\mathcal{H}$ de um código C é obtida a partir da matriz geradora $\mathcal{G}$ do código, sabendo que $\mathcal{H}\mathcal{G} = 0$, pois $\mathcal{H}$ gera $C^{\perp} = \{v \in K^n / \langle v, u \rangle = 0, \text{ para todo } u \in C\}$. Assim, se $\mathcal{G}$ é uma matriz $n \times k$, então $\mathcal{H}$ é uma matriz $(n-k) \times n$ de posto máximo $n-k$. Uma palavra $c \in C$ se, e somente se, existe x tal que $c = \mathcal{G}x$. Reescrevendo a equação de controle usando $\mathcal{H}$, temos $c \in C$ se, e somente se, $\mathcal{H}c = 0$.

Utilizando a matriz geradora na forma padrão, a matriz $\mathcal{H}$ na forma padrão pode ser facilmente obtida da seguinte forma: seja $\tilde{\mathcal{G}} = (I_k|A)^T$ a matriz geradora na forma padrão, a matriz de controle $\mathcal{H}$ na forma padrão é dada por $\tilde{\mathcal{H}} = (-A|I_{n-k})$.

De acordo com [8] temos a Cota de Singleton para códigos lineares apresentada pelo Corolário 1.2.18. Segundo [17], no contexto dos códigos de grupo à esquerda, temos, ainda, a cota de Singleton apresentada da seguinte maneira:

$$d \leqslant 1 + \sum_h n_h,$$

com n_h a dimensão do ideal à esquerda irredutível I_h e h percorre os índices correspondentes aos ideais minimais à esquerda que não aparecem na decomposição do ideal à

esquerda I que define o código.

4.4 Decodificação

Seja C um código de parâmetros (n, k, d). Considere $c \in C$ uma palavra transmitida e seja $r \in C$ a palavra recebida, isto é, a palavra c com um erro e ($r = c + e$) de peso menor que $\left[\frac{d-1}{2}\right]$.

O algoritmo de decodificação por distância mínima realiza a busca por uma palavra $c' \in C$ tal que a distância mínima desta palavra c' à palavra r seja mínima. Este procedimento pode ser descrito da seguinte maneira:

$$d(r, c') = min\{d(r, u)/u \in C\}.$$

A busca pela palavra de distância mínima em relação à palavra recebida pode ter um custo muito alto. A seguir será apresentado outro método de decodificação, conhecido como decodificação por síndrome, que visa a construção de algoritmos de decodificação de complexidade mais gerenciável.

4.4.1 Decodificação por síndrome

Seja $T = \{f_1, ..., f_N\}$ o conjunto de todos os geradores de ideais minimais em FG. Como as álgebras de grupo abordadas neste capítulo são semissimples, pode-se assumir que os seus geradores sejam idempotentes.

Seja y um gerador do código do grupo à esquerda $C = FGy$, que é a soma de m ideais minimais à esquerda, gerados por $f_{j_1}, ..., f_{j_m}$, isto é,

$$(FG)y = \bigoplus_{i=1}^{m} (FG)f_{j_i}$$

e os ideais minimais à esquerda $(FG)f_{j_i}$ são submódulos do ideal à esquerda gerado por

y, FGy, com suas interseções duas a duas sendo o ideal nulo.

Uma palavra-código de C tem a forma $c = xy$, com $x, y \in FG$. Como, cada elemento de $T\backslash\{f_{j_1}, ..., f_{j_m}\}$ é um anulador de y, temos

$$ct = 0, \quad \text{para todo } t \in T\backslash\{f_{j_1}, ..., f_{j_m}\}.$$

Além disso, como apresentado na Seção 4.3, $Hr^t = 0$ se, e somente se, r é um vetor do código. Deste modo, podemos definir o conjunto de $N - m$ síndromes da seguinte maneira:

$$S_t = rt = (c + e)t = ct + et = et, \quad \text{para todo } t \in T\backslash\{f_{j_1}, ..., f_{j_m}\}.$$

Decodificar por distância mínima é equivalente a procurar uma solução de peso mínimo de Hamming do conjunto de equações

$$\hat{e}t = S_t, \quad \text{para todo } t \in T\backslash\{f_{j_1}, ..., f_{j_m}\}.$$

Existe uma única palavra com o peso mínimo procurado se o número de erros for menor ou igual a $\left[\frac{d-1}{2}\right]$. Agora precisamos encontrar um algoritmo que decodifique com eficiência utilizando síndromes.

4.4.2 Correção por síndrome de um erro

Seja G um grupo não abeliano e F um corpo finito tal que $char\, K \nmid |G|$. Considere $f_1 = \sum_{g \in G} g$ o gerador do ideal minimal central de dimensão 1. Se quisermos usar o gerador idempotente, usaremos $e_1 = \frac{1}{|G|} f_1 = \frac{1}{|G|} \sum_{g \in G} g$.

Agora consideremos um código de grupo à esquerda $C = FGf$, de comprimento n, dimensão k, distância mínima $d \geqslant 3$. Vamos supor que o ideal FGf_1 não apareça na decomposição de FGf em ideais minimais. Isto significa $f_1 f = f f_1 = 0$. Como a distância

mínima de C é maior que 3 e pela cota de Singleton $d \leqslant 1 + \sum_h n_h$, então o código deve ter outros ideais anuladores irredutíveis com geradores $f_2, f_3, ..., f_t$.

Seja $c = xf$ a palavra transmitida e $r = c + e$ a palavra recebida com um único erro, dizemos que o padrão do erro é $e = \epsilon g_j$. Desta maneira calculamos t $(t \geqslant 2)$ síndromes:

$$S_i = rf_i, \qquad i = 1, ..., t,$$

com f_i os geradores dos ideais à esquerda irredutíveis anuladores de $C = FGf$. Procedendo desta maneira, para corrigir o erro necessitamos conhecer a posição j do erro e o módulo $\epsilon \in F$.

O Teorema 4.4.1 nos indica a capacidade corretora dos ideais à esquerda irredutíveis.

Teorema 4.4.1. *[17, Teorema 4.6] Seja G um grupo não abeliano tal que a característica p de F não divida a ordem do grupo. A distância mínima de qualquer código de grupo irredutível C, de FG de dimensão $k > 1$, é ao menos* 3.

Demonstração. Vamos primeiro observar que um código em FG pode ter distância mínima $d = |G|$, então os códigos irredutíveis de dimensão 1 podem ter distância mínima muito maior que 3.

Agora vamos supor um código FGy, cuja dimensão é maior que 1, isto é, este código tem mais de um vetor em sua base. Se a distância mínima for 2, existe uma palavra no código da forma $c = x_1 g_{j_1} + x_2 g_{j_2}$, com $g_{j_1} \neq g_{j_2}$ (note que não existem palavras no código com peso 1, ou seja, não há palavras no código da forma $x_1 g_j$). Logo, existe um elemento $x \in FG$ tal que

$$xy = x_1 g_{j_1} + x_2 g_{j_2}.$$

Como FGy é um ideal minimal, seja e_1 um gerador de outro ideal minimal, daí

$$xye_1 = 0.$$

$$(x_1 g_{j_1} + x_2 g_{j_2}) e_1 = 0$$

Como $g_{j_i} e_1 = e_1 = \frac{1}{|G|} \sum_{g \in G} g$, segue

$$(x_1 + x_2) e_1 = 0.$$

Assim, $x_1 + x_2 = 0$. Logo, reescrevendo a palavra de peso 2, obtemos

$$x_1 g_{j_1} + x_2 g_{j_2} = x_1 (g_{j_1} - g_{j_2}).$$

Podemos considerer uma representação fiel irredutível D_l tal que $D_l(y) = 0$. Assim, $0 = D_l(x) D_l(y) = D_l(xy) = D_l(x_1(g_{j_1} - g_{j_2})) = D_l(x_1).D_l(g_{j_1} - g_{j_2})$. Como a representação fiel é injetora e $x_1 \neq 0$, então $D_l(g_{j_1} - g_{j_2}) = 0$, o que implica $g_{j_1} = g_{j_2}$, visto que a representação é fiel, o que é uma contradição. □

No Teorema 4.4.2, veremos como podemos proceder na correção de um erro nos códigos de grupo à esquerda cujos parâmetros permitem corrigir pelo menos um erro. Neste caso, o erro $e = \epsilon g_j$, com j a posição em que o erro ocorreu e ϵ sua magnitude.

Perceba que, no caso de ideais de dimensão 1 e peso mínimo maior ou igual a 3, a correção de um erro é direta.

Teorema 4.4.2. *[17, Teorema 4.7] Seja $C = FGf$ um código à esquerda irredutível de dimensão maior que* 1*. Então*

1. *A magnitude do erro é obtida a partir da síndrome $S_1 = \epsilon f_1$.*
2. *O erro é localizado por meio de um algoritmo que simula a busca de Chien, usando as outras síndromes:*
   a) *Encontre os valores de i para os quais $\epsilon g_i f_2 - S_2$ é zero.*
   b) *Usa as síndromes restantes para selecionar a posição j.*

Demonstração. Seja $FG = FGf_1 \oplus FGf_2 \oplus ... \oplus FGf_r \oplus FGf_{r+1}$ a decomposição de FG em ideais à esquerda irredutíveis. Suponhamos $C = FGf = FG_{r+1}$, então $C^{\perp} = FGf_1 \oplus FGf_2 \oplus ... \oplus FGf_r$. Pelo Teorema 4.4.1, como a dimensão do código $C = FGf_{r+1}$ é maior que 1, por hipótese, então a distância mínima dele é no mínimo 3. Daí, pela Cota de Singleton, r é maior ou igual a 2.

Como estamos supondo que ocorreu somente um erro, então $e = \epsilon g_j$. Além disso, como $f_1 = \sum_{g \in G} g$, note que $g_j f_1 = f_1$, então $ef_1 = \epsilon g_j f_1 = \epsilon f_1$. Provando a parte 1, já que $S_1 = rf_1 = cf_1 + ef_1 = ef_1$.

Na parte 2 procuramos todos os valores de i para os quais $\epsilon g_i f_2 - S_2 = 0$. Note que $\epsilon g_j f_2 - S_2 = \epsilon g_j f_2 - rf_2 = \epsilon g_j f_2 - ef_2 = \epsilon g_j f_2 - \epsilon g_j f_2 = 0$. Assim, a posição do erro, j, sempre aparece no conjunto de índices calculado nessa etapa. Se apenas um índice j satisfizer a condição, já sabemos que o erro é ϵg_j.

Se o cálculo anterior resultar mais de um índice, então usamos as síndromes restantes para determinar a posição j do erro. De fato, j é o único índice tal que $\epsilon g_j f_h - S_h = 0$, para $h = 2, ..., r$, pois se houver outro índice l cumprindo $\epsilon g_l f_h - S_h = 0$, para $h = 2, ..., r$, então $\epsilon(g_l - g_j) f_h = 0,$ para $h = 2, ..., r$. Logo $g_l - g_j \in Ann\langle f_2, ..., f_r \rangle$.

Como $g_l - g_j \in Ann(f_1)$, já que $g_l f_1 = f_1 = g_j f_1$, então $g_l - g_j \in Ann\langle f_1, ..., f_r \rangle = I$, pois $\langle f_1, ..., f_r \rangle = I^{\perp}$, o que é uma contradição, pois $(g_l - g_j) \in I$ tem peso 2 e sabemos que o peso mínimo de I é 3. Portanto, somente um índice satisfaz $\epsilon g_j f_h - S_h = 0,$ para $h = 2, ..., r$. □

Observação: O Teorema 4.4.2 também pode ser aplicado no caso em que consideramos qualquer ideal à esquerda de peso mínimo maior ou igual a 3, tal que FGf_1 não apareça na decomposição de I como a soma de ideais minimais.

4.5 Um exemplo numérico

Considere o grupo simétrico S_3. Os elementos deste grupo

$$g_1 = (), \qquad g_2 = (1\,2\,3), \qquad g_3 = (1\,3\,2), \qquad g_4 = (1\,2), \qquad g_5 = (2\,3), \qquad g_6 = (1\,3),$$

formam as classes de conjugação

$$\begin{aligned} C_0 &= \{()\}; && \text{ordem } 1 \\ C_1 &= \{(1\,2\,3), (1\,3\,2)\}; && \text{ordem } 3 \\ C_2 &= \{(1\,2), (2\,3), (1\,3)\}; && \text{ordem } 2 \end{aligned}$$

com elementos de mesma ordem presentes na mesma classe.

Considerando a álgebra de grupo $A = F_5S_3$, temos 3 ideais centrais minimais gerados cada um por um idempotente primitivo central distinto, pois o número de idempotentes primitivos centrais é igual ao número de classes de conjugação de S_3. Observe os cálculos dos idempotentes primitivos centrais realizados no GAP.

```
gap> LoadPackage("wedderga");

gap> G:=SymmetricGroup(3);
Sym( [ 1 .. 3 ] )
gap> F:=GF(5);
GF(5)
gap> R:=GroupRing(F,G);
<algebra-with-one over GF(5), with 2 generators>

gap> d:=DirectSumDecomposition(R);
[ <two-sided ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>,
<two-sided ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>,
<two-sided ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)> ]

gap> GeneratorsOfIdeal(d[1]);
[ (Z(5)^0)*()+(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+
(Z(5)^0)*(1,3) ]
```

```
gap> GeneratorsOfIdeal(d[2]);
[ (Z(5)^0)*()+(Z(5)^2)*(2,3)+(Z(5)^2)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+
(Z(5)^2)*(1,3) ]
gap> GeneratorsOfIdeal(d[3]);
[ (Z(5)^2)*()+(Z(5)^3)*(1,2,3)+(Z(5)^3)*(1,3,2) ]
```

Estes idempotentes centrais também podem ser encontrados pela função:

```

idemp:=PrimitiveCentralIdempotentsByStrongSP(R);
[ (Z(5)^0)*()+(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+
(Z(5)^0)*(1,3),
(Z(5)^0)*()+(Z(5)^2)*(2,3)+(Z(5)^2)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+
(Z(5)^2)*(1,3),
(Z(5)^2)*()+(Z(5)^3)*(1,2,3)+(Z(5)^3)*(1,3,2) ]
```

Reescrevendo com base em

$$0 * Z(5) = 0, \quad , Z(5)^0 = 1, \quad Z(5) = 2, \quad Z(5)^2 = 4, \quad Z(5)^3 = 3$$

os idempotentes primitivos centrais, temos

$$\begin{aligned} e_1 &= g_1 + g_2 + g_3 + g_4 + g_5 + g_6 \\ e_2 &= g_1 + g_2 + g_3 - g_4 - g_5 - g_6 \\ e_3 &= 4g_1 - 2g_2 - 2g_3. \end{aligned}$$

Observe que

```
gap> e1:= idemp[1];
(Z(5)^0)*()+(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+(Z(5)^0)*(1,3)
gap> e2:= idemp[2];
(Z(5)^0)*()+(Z(5)^2)*(2,3)+(Z(5)^2)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+(Z(5)^2)*(1,3)
gap> e3:= idemp[3];
(Z(5)^2)*()+(Z(5)^3)*(1,2,3)+(Z(5)^3)*(1,3,2)

gap> e1*e2;
<zero> of ...
gap> e1*e3;
```

```
<zero> of ...
gap> e2*e3;
<zero> of ...
```

Isto é, $e_1e_2 = e_1e_3 = e_2e_3 = 0$.

Além disso, temos dois ideais minimais não centrais, gerados por

$$f_3 = 2g_1 + 3g_3 + 2g_4 + 3g_6,$$

$$f_4 = 2g_1 + 3g_2 + 3g_4 + 2g_6.$$

Observe que $e_3 = f_3 + f_4$, vide os cálculos feitos no GAP.

```
gap> f:=Embedding(G,R);
<mapping: SymmetricGroup( [ 1 .. 3 ] ) -> AlgebraWithOne( GF(5), ... ) >
gap> f3:=(Z(5))*()^f+(Z(5)^3)*(1,3,2)^f+(Z(5))*(1,2)^f+(Z(5)^3)*(1,3)^f;
(Z(5))*()+(Z(5))*(1,2)+(Z(5)^3)*(1,3,2)+(Z(5)^3)*(1,3)
gap> f4:=(Z(5))*()^f+(Z(5)^3)*(1,2,3)^f+(Z(5)^3)*(1,2)^f+(Z(5))*(1,3)^f;
(Z(5))*()+(Z(5)^3)*(1,2)+(Z(5)^3)*(1,2,3)+(Z(5))*(1,3)

gap> f3+f4=e3;
true

gap> idealf3+idealf4 = ideale3;
true

gap> Elements(Intersection(idealf3, idealf4));
[ <zero> of ... ]
```

Como o ideal gerado por e_3 é soma direta dos ideais gerados por f_3 e f_4, então ele é redutível.

Assim, temos dois códigos que são ideais à esquerda irredutíveis (gerados por f_3 e f_4), outros dois que são ideais bilaterais irredutíveis (gerados por e_1 e e_2) e mais um ideal minimal central, isto é, gerado por e_3, que é redutível.

Note que as contas no GAP envolvendo pemutações de grupos simétricos são realizadas da esquerda para a direita, como mostra o exemplo a seguir.

```
gap> g:=Elements(G);
[ (), (2,3), (1,2), (1,2,3), (1,3,2), (1,3) ]

gap> g[3]*g[4];
(1,3)
gap> g[4]*g[3];
(2,3)
```

Por isso devemos ter atenção ao calcular os ideais à esquerda.

Após esse processo calculamos os pesos das palavras desses códigos e suas respectivas dimensões.

```
gap> ideale1:=R*e1;
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>
gap> ideale2:=R*e2;
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>
gap> ideale3:=R*e3;
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>
gap> idealf3:=R*f3;
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>
gap> idealf4:=R*f4;
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>
gap> ideale1e2:=R*(e1+e2);
<left ideal in <algebra-with-one of dimension 6 over GF(5)>, (1 generators)>

gap> Dimension(ideale1);
1
gap> Dimension(ideale2);
1
gap> Dimension(ideale3);
4
gap> Dimension(idealf3);
2
gap> Dimension(idealf4);
2
```

```
gap> Dimension(ideale1e2);
2
```

Para calcular a distribuição de peso, foi utilizada a função *DistribuicaoDePeso* construída em [17] e já apresentada anteriormente nesse trabalho.

```

DistribuicaoDePeso:=function(I,R)
local wlist, k, j, dim, x, V, B, mf;
mf:=Size(LeftActingDomain(R))-1;
wlist:=List([0..Dimension(R)],x->0);
wlist[1]:=1;
dim:=Dimension(I);
B:=BasisVectors(Basis(I));
for j in [1..dim] do
    V:=SubspaceNC(R,B{[(j+1)..dim]});
    for x in V do
        k:=Size(CoefficientsAndMagmaElements(B[j]+x))/2+1;
        wlist[k]:=wlist[k]+mf;
    od;
od;
return wlist;
end;


gap> pesose1:=DistribuicaoDePeso(ideale1,R);
[ 1, 0, 0, 0, 0, 0, 4 ]
gap> pesose2:=DistribuicaoDePeso(ideale2,R);
[ 1, 0, 0, 0, 0, 0, 4 ]
gap> pesose3:=DistribuicaoDePeso(ideale3,R);
[ 1, 0, 24, 24, 144, 288, 144 ]
gap> pesosf3:=DistribuicaoDePeso(idealf3,R);
[ 1, 0, 0, 0, 12, 0, 12 ]
gap> pesosf4:=DistribuicaoDePeso(idealf4,R);
[ 1, 0, 0, 0, 12, 0, 12 ]
gap> pesose1e2:=DistribuicaoDePeso(ideale1e2,R);
[ 1, 0, 0, 8, 0, 0, 16 ]
```

Com base nos cálculos acima, temos a Tabela 4.1.

Tabela 4.1: Tabela de distribuição de pesos dos códigos

gerador	(n, k, d)	$w_H(x)$
e_1	(6, 1, 6)	$1 + 4x^6$
e_2	(6, 1, 6)	$1 + 4x^6$
e_3	(6, 4, 2)	$1 + 24x^2 + 24x^3 + 144x^4 + 288x^5 + 144x^6$
f_3	(6, 2, 4)	$1 + 12x^4 + 12x^6$
f_4	(6, 2, 4)	$1 + 12x^4 + 12x^6$
$e_1 + e_2$	(6, 2, 3)	$1 + 8x^3 + 16x^6$

Para estabelecer uma comparação, foi adicionado à tabela um gerador de um ideal não minimal $(e_1 + e_2)$. O resultado é que a distância mínima é pior com respeito aos ideais minimais de mesma dimensão.

4.5.1 Codificação e decodificação

Vamos reproduzir o processo realizado em [17] considerando o código gerado por f_3. Seus parâmetros são $[6, 2, 4]$, logo ele corrige um erro. Como é um código linear, o mesmo possui matriz geradora e matriz teste de paridade.

A matriz geradora é formada pelos vetores da base do código, daí calculamos essa base no GAP.

```
gap> BasisVectors(Basis(idealf3));
[ (Z(5)^0)*()+(Z(5)^0)*(1,2)+(Z(5)^2)*(1,3,2)+(Z(5)^2)*(1,3),
(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2,3)+(Z(5)^2)*(1,3,2)+(Z(5)^2)*(1,3) ]
```

Obtemos

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

Fazendo as contas, como $\mathcal{G}$ está na forma padrão $(I_k|A)^T$, então $\mathcal{H} = (-A|Id_{n-k})$. Logo,

$$\mathcal{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Os métodos algébricos padrões de códigos lineares utilizam as matrizes $\mathcal{G}$ e $\mathcal{H}$. Porém, iremos desenvolver este exemplo utilizando métodos próprios da estrutura de álgebra de grupo que não podem ser aplicados em outros tipos de códigos.

Considere $x = g_2 + g_3$ uma mensagem codificada como $c = xf_3 = 3g_1 + 2g_3 + 3g_4 + 2g_6$. Seja $r = 3g_1 + 2g_3 + 4g_4 + 2g_6$ a palavra recebida. Assim, podemos calcular três síndromes, que correspondem aos ideais minimais gerados por e_1, e_2 e f_4, isso porque $f_3 * e_1 = 0, f_3 * e_2 = 0, f_3 * e_3 \neq 0, f_3 * f_3 \neq 0$ e $f_3 * f_4 = 0$.

Lembre que é preciso ter cuidado ao fazer o cálculo de $r * f4$ no GAP, pois as contas no GAP envolvendo grupos simétricos são realizadas da esquerda para a direita. Logo, fazemos no GAP $f4 * r$. O mesmo não acontece com $r * e1$ e $r * e2$, pois $r * e1 = e1 * r$ e $r * e2 = e2 * r$.

```
1 gap> r := (Z(5)^3)*()^f + (Z(5))*(1,3,2)^f + (Z(5)^2)*(1,2)^f + (Z(5))*(1,3)^f;
2 (Z(5)^3)*()+(Z(5)^2)*(1,2)+(Z(5))*(1,3,2)+(Z(5))*(1,3)
3
```

```
gap> e1*r;
(Z(5)^0)*()+(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^0)*(1,2,3)+(Z(5)^0)*(1,3,2)+(Z(5)^0)*(1,3)
gap> e2*r;
(Z(5)^2)*()+(Z(5)^0)*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^2)*(1,2,3)+(Z(5)^2)*(1,3,2)+(Z(5)^0)*(1,3)
gap> f4*r;
(Z(5)^3)*()+(Z(5)^3)*(2,3)+(Z(5))*(1,2)+(Z(5))*(1,3,2)
```

Conforme os cálculos realizados, temos

$$S_1 = re_1 = e_1, \quad S_2 = re_2 = 4e_2, \quad S_3 = rf_4 = 3g_1 + 2g_3 + 2g_4 + 3g_5.$$

As síndromes são diferentes de zero, logo sabemos que ocorreram erros. Tentamos corrigí-los assumindo que é apenas um, ϵg_i. Se o procedimento falhar, significa que ocorreu mais de um erro e podemos detectá-los, entretanto não podemos corrigí-los, pois esse código corrige apenas um erro ($\kappa = (4-1)/2$).

A síndrome S_1 nos diz que a magnitude do erro é $\epsilon = 1$, pois $S_1 = re_1 = e_1$. A posição do erro pode ser encontrada pela "busca de Chien", ou seja,

$$\Delta = \epsilon g_i f_4 - S_3 = g_i(2g_1 + 3g_2 + 3g_4 + 2g_6) - (3g_1 + 2g_3 + 2g_4 + 3g_5)$$

é calculado para todo $i \in \{1, 2, 3, 4, 5, 6\}$. Os valores de i em que $\Delta = 0$ indicam possíveis posições de erro. Fazendo o cálculo para todos os valores de i obtemos os resultados da Tabela 4.2.

```
gap> delta1:=(f4*())-(f4*r);
(Z(5)^2)*()+(Z(5))*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^3)*(1,2,3)+(Z(5)^3)*(1,3,2)+(Z(5))*(1,3)

gap> delta2:=(f4*(1,2,3))-(f4*r);
(Z(5))*()+(Z(5)^2)*(2,3)+(Z(5)^3)*(1,2)+(Z(5))*(1,2,3)+(Z(5)^0)*(1,3,2)+(Z(5)^3)*(1,3)

gap> delta3:=(f4*(1,3,2))-(f4*r);
<zero> of ...

gap> delta4:=(f4*(1,2))-(f4*r);
```

```
<zero> of ...

gap> delta5:=(f4*(2,3))-(f4*r);
(Z(5))*()+(Z(5)^2)*(2,3)+(Z(5)^3)*(1,2)+(Z(5))*(1,2,3)+(Z(5)^0)*(1,3,2)+(Z(5)^3)*(1,3)

gap> delta6:=(f4*(1,3))-(f4*r);
(Z(5)^2)*()+(Z(5))*(2,3)+(Z(5)^0)*(1,2)+(Z(5)^3)*(1,2,3)+(Z(5)^3)*(1,3,2)+(Z(5))*(1,3)
```

Tabela 4.2: Cálculo do valor de $\Delta = \epsilon g_i f_4 - S_3$ para $i = 1, ..., 6$.

i	Δ
1	$4g_1 + 3g_2 + 3g_3 + g_4 + 2g_5 + 2g_6$
2	$2g_1 + 2g_2 + g_3 + 3g_4 + 4g_5 + 3g_6$
3	0
4	0
5	$2g_1 + 2g_2 + g_3 + 3g_4 + 4g_5 + 3g_6$
6	$4g_1 + 3g_2 + 3g_3 + g_4 + 2g_5 + 2g_6$

Desta tabela segue que o erro está na posição 3 ou 4. A ambiguidade se resolve usando a síndrome S_2 e calculando $g_3e_2 - S_2$ e $g_4e_2 - S_2$, daí

$$g_3e_2 - S_2 = 2g_1 + 2g_2 + 2g_3 + 3g_4 + 3g_5 + 3g_6$$

$$g_4e_2 - S_2 = 0.$$

Logo, houve um erro de magnitude 1 na posição 4 e daí $e = 0g_1 + 0g_2 + 0g_3 + 1g_4 + 0g_5 + 0g_6$. Além disso, como $e = r - c$, segue $c = r - e$. Assim,

$$c = 3g_1 + 2g_3 + 4g_4 + 2g_6 - (0g_1 + 0g_2 + 0g_3 + 1g_4 + 0g_5 + 0g_6).$$

Portanto,

$$c = 3g_1 + 2g_3 + 3g_4 + 2g_6.$$

Note que os códigos à esquerda gerados por f_3 e f_4 são melhores que os códigos gerados por e_1 e e_2 pois possuem dimensão maior, logo, tem mais palavras no código.

Lembramos aqui que os códigos à esquerda gerados por f_3 e f_4 também são melhores que o código gerado por $e_1 + e_2$, pois apesar de terem mesmo comprimento e mesma dimensão, este último apenas detecta dois erros e corrige apenas um erro, enquanto os gerados por f_3 e f_4 detectam três erros e corrigem até dois erros. Por último, comparamos os códigos à esquerda gerados por f_3 e f_4 com o código gerado por e_3. Novamente o primeiro é melhor que o segundo, pois o código gerado por e_3, apesar de ter uma dimensão maior, detecta apenas um erro e não corrige erros. Portanto, os códigos à esquerda gerados por f_3 e f_4 tem parâmetros melhores quando comparado com os outros códigos.

Nessa seção vimos a importância do estudo dos códigos não abelianos, uma vez que conseguimos um código não abeliano com parâmetros melhores que os códigos abelianos.

Considerações Finais

Nesse trabalho podemos perceber os benefícios dos estudos de códigos não abelianos na busca de códigos com parâmetros melhores. Analisamos também as limitações e dificuldades nos métodos de decodificação existentes. Assim, seguimos em busca de encontrar processos de decodificação mais eficientes.

Além disso, percebemos a importância das tecnologias para avançar nos estudos de códigos corretores de erros. Muito dos cálculos que aqui foram realizados utilizando a ferramenta GAP levariam anos para serem realizados por humanos.

Muito embora no trabalho não tenhamos explorado o contexto dos códigos de grupo no caso modular, foi necessário utilizar alguns resultados desse caso e por isso estudamos um pouco sobre o mesmo. Nesses estudos percebemos que o pacote "Wedderga" apenas funciona para o caso semissimples, para o caso modular o pacote utilizado chama-se "Laguna". Nesse contexto, percebemos um amplo escopo de pesquisa e possibilidades futuras.

Referências Bibliográficas

[1] ***GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra***. Manual GAP, 2009.

[2] BERNAL, J. J.; SIMÓN-PINERO, J. J.; DEL RÍO A. ***An intrinsical description of Group Codes***. Designs Codes and Cryptography, 2009.

[3] CRISTO, O. B.; KONOVALOV, A.; OLIVIERI A.; OLTEANU G.; DEL RÍO A. ***Wedderburn Decomposition of Group Algebras***. 2009.

[4] CURTIS, C. W.; REINER, I. ***Representation Theory of finite groups and associative algebras***. University of Oregon, University of Illinois, Interscience Publishers, a division of John Wiley & Sons, 1962.

[5] ELIA, M.; GORLA, E. ***On the computation of ideal dimensions in group algebras***. arXiv: 1403.7920.

[6] GUERREIRO, M.; MILIES, F. C. P. ***Group Algebras and Coding Theory***. Course Notes. CIMPA Research School on Algebraic Methods in Coding Theory, IME-USP, Julho 2017.

[7] HALL, M. ***The theory of groups***. Nova York: The Macmillan Company, 1959.

[8] HEFEZ, A.; VILLELA, M. L. T. ***Códigos Corretores de Erros***, *2ª edição*. Rio de Janeiro: Instituto Nacional de Matemática Pura e Aplicada - IMPA - Série de Computação e Matemática, 2008.

[9] HUNGERFORD, T. W. ***Graduate Texts in Mathematics - Algebra***. New York: Springer, 1974.

[10] JAMES, G.; KERBER, A. ***The Representation Theory of the Symmetric Group***. Encyclopedia of Mathematics and its Applications, Addison-Wesley Publishing Company, Reading Massachusetts, 1981.

[11] MACWILLIAMS, F. J.; SLOANE, N. J. A. ***The Theory of Error-Correcting Codes***, *Vol.16*. North-Holland: North-Holland Mathematical Library, 1977.

[12] MILIES, C. P. ***Breve introdução à Teoria dos Códigos Corretores de Erros***. IME-USP, Colóquio de Matemática da Região Centro-Oeste - SBM, 2009.

[13] MILIES, C. P. ***Anéis e Módulos***, *2ª edição*. Editora Livraria da Física, 2018.

[14] MILIES, C. P.; SEHGAL, S. K. ***An Introduction to Group Rings***. London: Algebras and Applications, Kluwer Academic Publishers, 2002.

[15] OLIVIERI, A.; DEL RÍO, A.; SIMÓN-PINERO J. J. S. ***On Monomial Characters and Central Idempotents of Rational Group Algebras***. Communications in Algebra, 2004.

[16] OLSHANSKY, A. Y. ***On the problem of numbers of generators and orders of abelian subgroups of finite p-groups***, *Vol.23, N.3*. Mat.Zametki, 1978.

[17] PILLADO, C. G. ***Códigos grupo no abelianos***. Oviedo: Universidade de Oviedo - Departamento de Matemática, 2015.

[18] PILLADO, C. G.; GONZÁLEZ, S.; MARKOV-V. T.; MARTÍNEZ C.; NECHAEV A. ***When are all group codes of a noncommutative group abelian (a computational approach)?***, *Vol.186, N.4*. Journal of Mathematical Sciences - Springer Science+Business Media New York, 2012.