UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
LICENCIATURA EM MATEMÁTICA

DÉBORA AMANDA MARQUEZ

CÓDIGOS SOBRE GRUPOS E LOOPS

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2018

DÉBORA AMANDA MARQUEZ

CÓDIGOS SOBRE GRUPOS E LOOPS

Trabalho de Conclusão de Curso apresentada ao curso de Licenciatura em Matemática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Licenciado em Matemática".

Orientador: Tiago Henrique dos Reis

CORNÉLIO PROCÓPIO

2018

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação
Departamento de Matemática
Curso de Licenciatura em Matemática

FOLHA DE APROVAÇÃO

BANCA EXAMINADORA

_____

Prof. Tiago Henrique dos Reis
(Orientador)

_____

Prof. Tiago Debarba

_____

Prof. Josimar da Silva Rocha

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso"

AGRADECIMENTOS

Agradeço à minha mãe, Jovelina Alves Marquez, que me apoiou durante toda a minha graduação, tornando isso tudo possível. Ao meu pai João Marquez (*in memoriam*), que sempre foi minha maior fonte de inspiração e força.

Sou grata ao meu orientador, Tiago Henrique dos Reis, pela paciência, dedicação, e por nunca ter desistido de mim. E acima de tudo, pelo incentivo, pois muitas vezes foi o empurrão que precisava.

À banca examinadora, prof. Tiago Debarba, e o prof. Josimar da Silva Rocha, pelas essenciais contribuições apresentadas para esse trabalho.

E, por fim, obrigado à todos os meus amigos em especial Mirian F. D. G. Correia e meu namorado Ricardo Muniz da Silva que fizeram parte de tudo isso.

RESUMO

MARQUEZ, D. A.. CÓDIGOS SOBRE GRUPOS E LOOPS. 43 f. Trabalho de Conclusão de Curso – Licenciatura em Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

Este trabalho apresenta uma análise sobre Códigos de Verificação de Erros, mostrando alguns conceitos que a fundamentam, como a Teoria dos Números, Grupos e Loops. Também será discutido a importância dos códigos na Teoria de Informação. Analisaremos alguns sistemas de verificação em relação a capacidade de detecção de erros, conforme Verhoeff, propondo também um novo sistema de verificação de erros, utilizando Loops.

Palavras-chave: Teoria de Códigos, Verificação de Erro, Álgebra

ABSTRACT

MARQUEZ, D. A.. TITLE IN ENGLISH. 43 f. Trabalho de Conclusão de Curso – Licenciatura em Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2018.

This paper presents an analysis on Error Checking Codes, discussing some concepts that support it, such as the Number Theory, Groups and Loops. The importance of codes in Information Theory will also be presented. Checking systems in relation to the error detection capability will be annalyzed, according to Verhoeff, also proposing a new error checking system, using Loops.

Keywords: Codes Theory, Error Checking, Algebra

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Um sistema de comunicação é um conjunto de mecanismos da qual fluem as informações, permitindo processá-las e transmiti-las desde a origem (emissor) até o destino (receptor). Sempre que é transmitida uma mensagem, tem-se o risco de ocorrer alguns problemas, como erros nos equipamentos, interferências ou até mesmo erros de digitação no lançamento de informação. Este último tipo de erro em especial é o foco deste trabalho. Ruído é o nome dado para qualquer tipo de erro.

A transmissão de mensagens tem dois principais problemas associados, detectar e corrigir erros ocorridos e a segurança nessa transmissão. O interesse deste trabalho é estudar os meios para detectar falhas, assegurar a confiabilidade na comunicação sob a perspectiva matemática. Uma vez que, dada mensagem original, utiliza-se a adição de mais informações redundantes, chamados de dígitos verificadores, que são obtidos através de operações matemáticas, para detectar erros de transmissão.

Todos os meios de comunicação e sistemas de identificação, como por exemplo o cartão de crédito, carteira de identidade, certidão de nascimento/casamento/óbito, CNPJ, CPF, matrícula de servidores, notas fiscais, título de eleitor, códigos de barras, entre outros, utilizam um mecanismo de verificação de erro.

O objetivo deste trabalho é estudar fundamentos da Teoria de Códigos Verificadores de Erros, compreendendo os conceitos matemáticos que dão suporte para tal teoria, estudar a eficiência de alguns códigos e propor um código de verificação. No Referencial Teórico, será apresentado alguns conceitos de Teoria de Números (teorema fundamental da aritmética e aritmética modular), Teoria de Grupos (grupos de permutação e grupos diedrais) e Loops, utilizando como referências os livros (MILIES; COELHO, 2001), (DOMINGUES; IEZZI, 2003) e (KANDASAMY, 2002).

No capítulo 3, serão apresentados alguns sistemas de verificação e também será feita uma análise preliminar de tais sistemas, tendo como base os artigos (BROWN, 1974; LARSEN, 1983; MILIES; MILIES, 2013) e o capítulo 12 do livro (SÁ; ROCHA, 2010). Em especial,

destacamos a tabela com a frequência relativa de erros mais comuns cometidos por operadores humanos apresentada pelo matemático Jacobus Koos Verhoeff e as consequências de tal análise para a teoria.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta algumas definições e resultados que serão utilizados neste trabalho. Optamos por omitir as demonstrações da maioria dos teoremas afim de explorar os resultados, para aplicar no decorrer do nosso trabalho. As demonstrações dos resultados desse capítulo podem ser encontradas em (MILIES; COELHO, 2001) e (DOMINGUES; IEZZI, 2003), no que se refere a Teoria dos Números e a Teoria de Grupos, respectivamente.

2.1 TEORIA DE NÚMEROS

A Teoria dos Números é a área da matemática que estuda o anel dos números inteiros, denotado por $\mathbb{Z}$, com duas operações: a adição e a multiplicação. Esta teoria pode ser subdividida em vários campos, de acordo com os métodos usados e das questões investigadas. Por simplicidade, iremos omitir algumas demonstrações da construção de $\mathbb{Z}$ e suas propriedades mais elementares.

2.1.1 TEOREMA FUNDAMENTAL DA ARITMÉTICA

Dados $a, b \in \mathbb{Z}$, uma equação do tipo $bx = a$ pode ou não ter solução no conjunto dos números inteiros; isso dependerá dos coeficientes a e b da equação. Vejamos um exemplo:

Exemplo 1. Para $a = 6$ e $b = 2$, temos: $2 \cdot x = 6$ com $x = 3$

Por outro lado para $a = 11$ e $b = 5$, temos: $5 \cdot x = 11$ com $x \notin \mathbb{Z}$

Quando tal solução existe diz-se que a é *divisível* por b, como acontece no primeiro caso.

Definição 1. Sejam $a, b \in \mathbb{Z}$. Diz-se que b divide a se existe um inteiro $c \neq 0$ tal que $bc = a$.

Usaremos a notação $b \mid a$ para indicar que b divide a. A negação dessa afirmação será indicada por $b \nmid a$. A próxima proposição, reúne algumas propriedades elementares da divisibilidade.

Proposição 1. Para todo $a,b,c,d \in \mathbb{Z}$ vale:

(i) $a \mid a$.

(ii) Se $a \mid b$ e $b \mid c$, então $a \mid c$.

(iii) Se $a \mid b$ e $c \mid d$, então $ac \mid bd$.

(iv) Se $a \mid b$ e $a \mid c$, então $a \mid (b+c)$.

(v) Se $a \mid b$ então $\forall\, m \in \mathbb{Z}$, tem-se que $a \mid mb$.

(vi) Se $a \mid b$ e $a \mid c$, então, $\forall\, m,n \in \mathbb{Z}$, tem-se que $a \mid (mb+nc)$.

Lema 1. Sejam a e b inteiros, tais que $a \geq 0$ e $b > 0$. Então, existem q e r, tais que $a = bq + r$ e $0 \leq r < b$.

Os números q e r determinados no lema anterior chamam-se, respectivamente, quociente e resto da divisão de a por b.

Demonstração. Consideremos o seguinte conjunto

$$S = \{a - bx | x \in \mathbb{Z}, a - bx \geq 0\}$$

Quando $x = 0$, temos que $a - bx = a \geq 0$ é um elemento de S. Logo, $S \neq \emptyset$.

Pelo Princípio da Boa Ordem, existe $r = mim\ S$. Como $r \in S$, ele também é da forma $r = a - bq \geq 0$, para algum $q \in \mathbb{Z}$.

Para mostrar que as condições do enunciado estão verificadas, bastará provar que $r < b$. De fato, se fosse $r \geq b$, teríamos que:

$$a - b(q+1) = a - bq - b + r - b \geq 0,$$

Logo, $a - b\ (q+1)$ também pertenceria a S.

Mas $a - b\ (q+1) = r - b < r = mim\ S$, uma contradição. □

A seguir apresentaremos o algoritmo da divisão. Este teorema é importante, pois garante que dado dois números inteiros a e b com $b \neq 0$ a divisão de a por b é possível, se for deixado um resto que respeita certas condições. Este resto é a base para a construção de toda

aritmética modular, e a unicidade do par quociente-resto é fundamental para tanto. Note que, de certa forma, este teorema é uma generalização da Definição 1.

Teorema 1 (Algoritmo da Divisão)**.** Sejam $a, b \in \mathbb{Z}$, com $b \neq 0$. Então, existem q e r, únicos, tais que:

$$a = bq + r \text{ e } 0 \leq r < \mid b \mid .$$

A demostração segue da aplicação do lema para os 4 casos, $a \geq 0$ e $b > 0$, $a < 0$ e $b > 0$, $a < 0$ e $b < 0$, $a \geq 0$ e $b < 0$, onde o primeiro caso é o próprio lema 1.

Exemplo 2.

- Para $a = 5$ e $b = 3$ temos que a divisão de a por b é dada por:

$$5 = 3 \cdot 1 + 2 \text{ com } q = 1 \text{ e } r = 2.$$

- Para $a = 5$ e $b = -3$ temos que a divisão de a por b é dada por:

$$5 = -3 \cdot (-1) + 2 \text{ com } q = -1 \text{ e } r = 2.$$

- Para $a = -100$ e $b = -7$ temos que a divisão de a por b é dada por:

$$-100 = -7 \cdot 15 + 5 \text{ com } q = 15 \text{ e } r = 5.$$

- Para $a = -100$ e $b = 7$ temos que a divisão de a por b é dada por:

$$-100 = 7 \cdot -15 + 5 \text{ com } q = -15 \text{ e } r = 5.$$

O próximo teorema será usado para demostrar uma propriedade importante da seção 2.2.

Lema 2 (Teorema de Bezout)**.** Sejam a, b inteiros e $d = \text{mdc}(a, b)$. Então, existem inteiros r e s tais que $d = ra + sb$.

O principal objetivo deste capítulo é apresentar o Teorema Fundamental da Aritmética, mas para isso é interessante conhecer um pouco sobre os números primos. Desde a antiguidade, os números primos são estudados por vários matemáticos nomeados, sendo alguns deles Euclides, Diofanto, Eratóstenes; o primeiro a criar uma tabela de números primos (crivo de Eratóstenes), entre outros. (BOYER; MERZBACH, 1996).

Um número primo é um inteiro que possui dois divisores positivos, 1 e seu módulo. Existe um especial interesse nos números primos pelo fato de que todo número maior do que 1

ou é primo ou é escrito como produto de números primos de modo único (exceto pela ordem dos fatores). Quando isso acontece diz-se que este número é decomposto.

Definição 2. Um inteiro p é primo se tem exatamente dois divisores positivos, 1 e $\mid p \mid$.

A decomposição tem grande importância no Sistema de Verificação de Erros, como veremos no próximo capítulo. Isso também abre diversas possibilidades de aplicações, como em criptografia, onde uma mensagem pode ser codificado utilizando números primos. Também é importante na Teoria dos Números, servindo como base para provar diversos resultados.

Apresentamos o teorema fundamental da aritmética que diz que todo número inteiro diferente de 0, 1 e -1 ou é primo, ou pode ser escrito como produto de primos. Por exemplo, 150 é decomposto como $2 \cdot 3 \cdot 5^2$.

Teorema 2 (Teorema Fundamental da Aritmética)**.** Seja a um inteiro diferente de $0, 1$ e -1. Então, existem primos positivos $p_1 < p_2 < \cdots < p_r$ e inteiros positivos $n_1, n_2, \ldots, n_r$ tais que $a = \pm p_1^{n_1} p_2^{n_2} \ldots p_r^{n_r}$. Além disso, essa decomposição é única.

Apesar do Teorema Fundamental da Aritmética garantir a existência da decomposição e da unicidade, nada diz sobre como obter tal decomposição. Existem alguns métodos, como divisões sucessivas (minimo múltiplo comum), fatoração por Fermat e árvore de fatores, para obter tal decomposição. As propriedades dos números primos são utilizadas como base pela criptografia, por exemplo. Na Teoria de Código Verificador de Erros, utiliza-se principalmente o fato de que um número primo não possui divisores positivos, com exceção de um e ele mesmo.

2.1.2 ARITMÉTICA MODULAR

Segundo MELO (2015, p. 14), "Johann Carl Friedrich Gauss, matemático alemão, desenvolveu feitos em várias áreas da ciência, em particular na utilização e desenvolvimento da aritmética modular", a qual faremos uma breve discussão, sendo considerada como um dos conceitos mais fortes da Teoria dos Números. A congruência também conhecida como aritmética do relógio, é uma relação entre os números a e b inteiros, dada pelo resto da divisão desses números por um outro número m fixado.

Definição 3. Seja $m > 0$ um número fixo. Dois inteiros a e b dizem-se congruentes módulo m se m divide a diferença $a - b$.

Utiliza-se a notação $a \equiv b \pmod{m}$.

Observando o funcionamento de um relógio, é possível notar que depois do 12 os números voltam a se repetir, por exemplo a posição do ponteiro quando passado 3 horas a partir do meio dia, ou seja $12+3$, diferente da aritmética usual em que $12+3=15$, na aritmética do relógio o ponteiro estará marcando 3 horas, pois $12+3=3$ equivale a 15. Assim, o conjunto de todos os números equivalentes ao número 3 é $\overline{3}=\{3+12k;\ k\in\mathbb{Z}\}$.

A próxima proposição permite estudar congruência a partir do resto de dois inteiros quando divididos pelo módulo como discutido inicialmente. Alguns livros trazem esta propriedade como definição. Em outras palavras, admite que dois números são congruentes se, e só se tem o mesmo resto quando divididos por m, e prova como propriedade que se a diferença de dois inteiros for divisível por m, então estes são congruentes.

Proposição 2. Seja m um número fixo. Dois inteiros a e b são congruêntes módulo m se e somente se eles têm como resto o mesmo inteiro quando dividimos por m.

Demonstração. Sem perda de generalidade, suponha $r_1 > r_2$.

$$a = mq_1 + r_1, \text{com} \quad 0 \leq r_1 < m$$

$$b = mq_2 + r_2, \text{com} \quad 0 \leq r_2 < m.$$

Então,

$$a - b = m\,(q_1 - q_2) + (r_1 - r_2),$$

Logo,

$$m \mid (a-b) \text{ se e somente se } m \mid (r_1 - r_2).$$

Ainda, como $0 \leq |r_1 - r_2| < m$, temos que $m \mid (r_1 - r_2)$ se e somente se $r_1 - r_2 = 0$.

Consequentemente, $a \equiv b \pmod{m}$ se e somente se $r_1 = r_2$. □

Exemplo 3. Se hoje é quarta-feira, que dia será daqui a 3842 dias? Sabendo que uma semana tem 7 dias começando por quarta-feira temos que $3842 = 7 \cdot 548 + 6$, ou ainda, $3842 \equiv 6 \mod 7$. Logo, se hoje é quarta-feira, daqui 3842 dias, ou seja daqui 548 semanas e 6 dias será terça-feira.

Exemplo 4. Seja $m = 3$, $a = 10$ e $b = 7$. Pela Definição 3, se $10 \equiv 7 \pmod 3$ então $3 \mid 10 - 7$. De fato, $10 - 7 = 3$ e $3 \mid 3$. Por outro lado, note que $10 = 3 \cdot 3 + 1$ e $7 = 3 \cdot 2 + 1$ ou seja, a e b possuem o mesmo resto quando divididos por 3. Portanto, $10 \equiv 7 \pmod 3$ usando a proposição 2.

Similar a Proposição 1, o próximo resultado traz algumas propriedades básicas de congruência. A demonstração segue dos resultados da Proposição 1.

Proposição 3. Sejam $m > 0$ um número fixo, e a, b, c, d inteiros arbitrários. Então, valem as seguintes propriedades:

i) $a \equiv a \pmod{m}$.

ii) Se $a \equiv b \pmod{m}$, então $b \equiv a \pmod{m}$.

iii) Se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $a \equiv c \pmod{m}$.

iv) Se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, então $a + c \equiv b + d \pmod{m}$.

v) Se $a \equiv b \pmod{m}$, então $a + c \equiv b + c \pmod{m}$.

vi) Se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, então $ac \equiv bd \pmod{m}$.

vii) Se $a \equiv b \pmod{m}$, então $a^n \equiv b^n \pmod{m}$ para todo inteiro positivo n.

viii) Se $a + c \equiv b + c \pmod{m}$, então $a \equiv b \pmod{m}$.

Definição 4. Seja $a \in \mathbb{Z}$, chama-se classe de congruência de a módulo m o conjunto formado por todos os inteiros que são congruêntes a a módulo m.

Notação: $\overline{a} = \{x \in \mathbb{Z} \mid x \equiv a \pmod{m}\} = \{a + tm \mid t \in \mathbb{Z}\}$.

Exemplo 5. Seja $a = 4$ e $m = 5$ temos então que a classe de congruência de a modulo m é;

$$\overline{4} = \{x \in \mathbb{Z} \mid x \equiv 4 \pmod{5}\} = \{4 + t5 \mid t \in \mathbb{Z}\} = \{\cdots, -11, -6, -1, 4, 9, 14, 19, \cdots\}.$$

Considere agora $a = 2$ para o mesmo m, temos então que a classe de congruência de a modulo m é;

$$\overline{2} = \{x \in \mathbb{Z} \mid x \equiv 2 \pmod{5}\} = \{2 + t5 \mid t \in \mathbb{Z}\} = \{\cdots, -13, -8, -3, 2, 7, 12, 17, ...\}.$$

Proposição 4. Sejam a e b inteiros. Então $a \equiv b \pmod{m}$ se e somente se $\overline{a} = \overline{b}$.

Demonstração. Suponhamos que $a \equiv b \pmod{m}$: queremos provar que $\overline{a} = \overline{b}$, isto é, uma igualdade entre conjuntos.

Dado $x \in \overline{a}$, temos por definição, que $x \equiv a \pmod{m}$. Da propriedade transitiva de congruência e da hipótese, segue imediatamente que $x \in \overline{b}$. Logo, $\overline{a} \subset \overline{b}$. A inclusão de sentido contrário segue de forma análoga.

Reciprocamente, se $\overline{a} = \overline{b}$, como $a \in \overline{a}$, temos também que $a \in \overline{b}$, logo, $a \equiv b \pmod{m}$.

□

Fixado m, o conjunto de todas as classes módulo m será de especial interesse para nós e será denotado por $\mathbb{Z}_m$.

Exemplo 6. $\mathbb{Z}_5 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \}$ os representantes das classes são $0, 1, 2, 3, 4$ como vimos, cada classe representa os restos da divisão deles por m.

Nosso objetivo é fixado um valor para m, estudar o conjunto $\mathbb{Z}_m$, com duas operações fechadas, induzidas a partir das operações usuais nos inteiros. Essas operações são obtidas a partir da soma e produto usual nos inteiros.

$$+ \colon \mathbb{Z}_m \times \mathbb{Z}_m \longrightarrow \mathbb{Z}_m \qquad\qquad \cdot \colon \mathbb{Z}_m \times \mathbb{Z}_m \longrightarrow \mathbb{Z}_m$$
$$\overline{a} + \overline{b} \longmapsto \overline{a+b} \in \mathbb{Z}_m \qquad\qquad \overline{a} \cdot \overline{b} \longmapsto \overline{a \cdot b} \in \mathbb{Z}_m$$

Os próximos resultados mostram que as operações não dependem do representante que se toma de uma classe, e também que são válidas algumas das propriedades elementares do conjunto dos números inteiros.

Lema 3. Sejam a, a', b e b' inteiros tais que $\overline{a} = \overline{a'}$ e $\overline{b} = \overline{b'}$. Então, $\overline{a+b} = \overline{a'+b'}$ e $\overline{ab} = \overline{a'b'}$.

Proposição 5. Em $\mathbb{Z}_m$ valem as seguintes propriedades:

Associatividade da Soma: Para toda terna $\overline{a}$, $\overline{b}$, $\overline{c}$ de inteiros módulo m, tem-se que $\overline{a} + (\overline{b} + \overline{c}) = (\overline{a} + \overline{b}) + \overline{c}$.

Existência do Neutro da Soma: Existem um único elemento em $\mathbb{Z}_m$, que é precisamente $\overline{0}$, tal que $\overline{a} + \overline{0} = \overline{a} \ \ \forall \ \ \overline{a} \in \mathbb{Z}_m$.

Existência do oposto da Soma: Para cada inteiro módulo m, $\overline{a}$, existe um único elemento em $\mathbb{Z}_m$, que chamaremos oposto de $\overline{a}$ e indicaremos por $\overline{-a}$, tal que $\overline{a} + \overline{-a} = \overline{0}$.

Comutatividade da Soma: Para todo par $\overline{a}$, $\overline{b}$ de $\mathbb{Z}_m$, tem-se que $\overline{a} + \overline{b} = \overline{b} + \overline{a}$.

Associatividade do Produto: Para toda terna $\overline{a}$, $\overline{b}$, $\overline{c}$ de inteiros módulo m, tem-se que $\overline{a} \cdot (\overline{b} \cdot \overline{c}) = (\overline{a} \cdot \overline{b}) \cdot \overline{c}$.

Existência do Neutro do Produto: Existem um único elemento em $\mathbb{Z}_m$, que é precisamente $\overline{1}$, tal que $\overline{a} \cdot \overline{1} = \overline{a} \ \ \forall \ \ \overline{a} \in \mathbb{Z}_m^*$.

Comutatividade do Produto: Para todo par $\overline{a}$, $\overline{b}$ de $\mathbb{Z}_m$, tem-se que $\overline{a} \cdot \overline{b} = \overline{b} \cdot \overline{a}$.

Distributividade: Para toda terna $\overline{a}$, $\overline{b}$, $\overline{c}$ de inteiros módulo m, tem-se que $\overline{a} \cdot (\overline{b} + \overline{c}) = \overline{a} \cdot \overline{b} + \overline{a} \cdot \overline{c}$.

A demonstração dessa proposição segue diretamente das propriedades elementares dos inteiros. As próximas definições e resultados apresentam algumas propriedades de $\mathbb{Z}_m$ que serão bastante úteis na construção dos sistemas de verificação.

Definição 5. Um elemento $\overline{a} \in \mathbb{Z}_m$ diz-se inversível se existe $\overline{a'} \in \mathbb{Z}_m$ tal que $\overline{a} \cdot \overline{a'} = \overline{1}$. Um elemento $\overline{a'}$ nessas condições diz-se um inverso de $\overline{a}$. Para esse mesmo $\overline{a}$ não nulo, diz-se um divisor de zero se existe $\overline{b} \in \mathbb{Z}_m$, também não nulo, tal que $\overline{a} \cdot \overline{b} = \overline{0}$.

Proposição 6. Um elemento não nulo $\overline{a}$ de $\mathbb{Z}_m$ é divisor de zero se, e somente se, o máximo divisor comum entre a e m é diferente de 1 ($\text{mdc}(a,m) \neq 1$). Reciprocamente, $\overline{a} \in \mathbb{Z}_m$ é inversível se, e somente se, o máximo divisor comum entre a e m é igual a 1 ($\text{mdc}(a,m) = 1$).

Note que, na proposição 5 não listamos a propriedades cancelativa, tal que para todo $\overline{a} \neq \overline{0}, \overline{b}, \overline{c} \in \mathbb{Z}_m$, vale $\overline{b} \cdot \overline{a} = \overline{c} \cdot \overline{a} \Rightarrow \overline{b} = \overline{c}$, pois nem sempre em $\mathbb{Z}_m$ essa propriedade vale.

Lema 4. Se $\mathbb{Z}_m$ não contém divisores de zero, então m é um inteiro primo, e a propriedade cancelativa do produto vale em $\mathbb{Z}_m$ se e somente se m é primo.

Corolário 1. Seja $p > 1$ um inteiro primo. Então, $\mathbb{Z}_p$ não contém divisores de zero e todo elemento não nulo é inversível.

2.2 GRUPOS

Joseph Louis Lagrange (1736 - 1813), matemático e físico contribuiu significativamente em "várias áreas da ciência, dentre eles: a teoria dos números; teoria das funções; cálculo de probabilidades; teoria dos grupos; equações diferenciais; mecânica dos fluidos; mecânica analítica e mecânica celeste" (USP, 2012, online). Lagrange reconheceu a importância da teoria das permutações para a resolução de equações. Visto que, uma dificuldade era a determinação de raízes para polinômios de grau maior ou igual a 5. Niels Henrik Abel (1802 - 1829) em 1824, provou que nem todas as equações polinomiais de grau maior ou igual a 5, admitiam métodos de solução por radicais. A partir deste estudo houve a primeira introdução do conceito matemático de grupo. (DOMINGUES; IEZZI, 2003).

Evariste Galois (1811 - 1832), autor do conceito de grupo e o primeiro a considerar explicitamente grupos de permutações, associando a cada equação um grupo de permutações. De acordo com Malsev (apud ALEKSANDROV; KOLMOGOROV, 1994, P. 310) a Teoria dos Grupos nasceu da necessidade de encontrar um método para estudar propriedades importantes do mundo real, tal como a simetria, presentes no cotidiano e na natureza como por exemplo, nas asas de uma borboleta, as pétalas de uma flor ou uma concha do mar, e sua importância na

resoluções de equações matemáticas. Possuindo consequências na Teoria das Equações, Teoria dos Números, Geometria Diferencial, Cristalografia.

Definição 6. Um sistema matemático constituído de um conjunto não vazi o G e uma operação fechada $(x,y) \mapsto x * y$ sobre G é chamado *grupo* se essa operação se sujeita aos seguintes axiomas:

(i) *associatividade*:

$(a*b)*c = a*(b*c)$, quaisquer que sejam $a,b,c \in G$;

(ii) *existência de elemento neutro*:

Existe um elemento $e \in G$ tal que $a*e = e*a = a$, qualquer que seja $a \in G$;

(iii) *existência de simétricos*:

Para todo $a \in G$ existe um elemento $a' \in G$ tal que $a*a' = a'*a = e$.

Niels Henrik Abel também possui influência sobre a origem da Teoria de Grupos, devido seus estudos sobre grupos comutativos, por este motivo, esses grupos são também denominados por grupos abelianos. Segundo PINTO (2009, p. 2) "Os grupos abelianos possuem importância central em Álgebra Abstrata e em outros ramos da Matemática especialmente na Topologia Algébrica". Se, além dos axiomas supracitados, ainda se cumprir o axioma da *comutatividade* $a*b = b*a$, quaisquer que seja $a,b \in G$, o grupo recebe o nome de *grupo comutativo ou abeliano*.

Exemplo 7. Sistema formado pelo conjunto dos inteiros e a adição usual, denotado por $(\mathbb{Z},+)$ é um grupo. A adição usual é uma operação sobre $\mathbb{Z}$, associativa e comutativa. Há um elemento neutro para ela (o número 0), e o oposto $-a$ de um elemento $a \in \mathbb{Z}$ também pertence a esse conjunto. Pelo mesmo motivo $(\mathbb{Q},+)$, $(\mathbb{R},+)$ e $(\mathbb{C},+)$ são grupos.

Por outro lado $(\mathbb{Z}^*,\cdot)$ não é grupo, pois o sistema formado pelo conjunto $\mathbb{Z}^*$ e a multiplicação embora o produto de dois inteiros não nulos seja sempre um inteiro não nulo, ocorre que nenhum inteiro a, salvo 1 e -1, tem inverso em $\mathbb{Z}$.

Sistema formado pelo conjunto dos racionais não nulos e a multiplicação usual sobre esse conjunto, denotado por $(\mathbb{Q}^*,\cdot)$, também é um grupo. O conjunto $\mathbb{Q}^*$ é formado em relação à multiplicação, ou seja, o produto de dois números racionais não nulos também é diferente de zero. A multiplicação usual é associativa em $\mathbb{Q}^*$ porque é em $\mathbb{Q}$, e o número 1 é elemento neutro da multiplicação. O inverso de um elemento a é a^{-1}. Pelo mesmo motivo, $(\mathbb{R},\cdot)$ e $(\mathbb{C}^*,\cdot)$ também são grupos.

$(\mathbb{Z}_m, +)$ também é grupo. Este resultado segue diretamente da proposição 5.

Propriedades imediatas de um grupo

Seja $(G, *)$ um grupo. As propriedades para uma operação sobre um conjunto nos asseguram a unicidade de elemento neutro, do simétrico de cada elemento de G, além de que, se e é o elemento neutro, então $e' = e$ do mesmo modo $(a')' = a$, qualquer que seja $a \in G$. Note ainda que $(a * b)' = b' * a'$ e, por indução, $(a_1 * a_2 * \cdots * a_n)' = a'_n * a'_{n-1} * \cdots * a'_1 (n \geq 1)$. Todo elemento de G é regular para a operação $*$, ou seja; se $a * x = a * y$, então $x = y$.

Teorema 3. Seja G um grupo, então a equação $a * x = b \quad (x * a = b)$ tem conjunto solução unitário, constituído do elemento $a' * b$ (respectivamente $b * a'$).

Demonstração. Consideremos $a * x = b$. Substituindo x por $a' * b$ no primeiro membro da equação, obtém-se

$$a * (a' * b) = (a * a') * b = e * b = b.$$

o que garante que efetivamente $a' * b$ é solução da equação. Por outro lado, suponha que x_0 também é solução da equação, então $a * x_0 = b$. Dai:

$$a' * (a * x_0) = a' * b.$$

Como $a' * (a * x_0) = (a' * a) * x_0 = x_0$, então $x_0 = a' * b = x$.

Portanto, $a * x = b$ é solução única. □

Exemplo 8. $(\mathbb{Z}_p^*, \cdot)$ é grupo Abeliano.

De fato, vimos na proposição 5 as propriedades válidas em $\mathbb{Z}_m$ na multiplicação: Associativa, comutativa e existência do neutro. Para que $\mathbb{Z}_p^*$ seja um grupo basta verificar que vale a restrição da multiplicação e existência do simétrico.

Teorema 4. A restrição da multiplicação módulo m aos elementos de $\mathbb{Z}_m^*$ é uma operação sobre esse conjunto se, e somente se m é primo.

Teorema 5. Qualquer que seja o elemento $\overline{a} \in \mathbb{Z}_m^*$, pode-se encontrar $\overline{b} \in \mathbb{Z}_m^*$ tal que $\overline{a} \cdot \overline{b} = 1$.

Demonstração. De fato, se $\overline{a} \in \mathbb{Z}_m^*$, então a não é múltiplo de m. Como m é primo, então $\text{mdc}(m, a) = 1$. Pelo teorema 2, temos que $mx_0 + ay_0 = 1$, para convenientes inteiros x_0 e y_0. Reduzindo-se essa igualdade, módulo m:

$\overline{mx_0 + ay_0} = \overline{m} \cdot \overline{x_0} + \overline{a} \cdot \overline{y_0} = \overline{a} \cdot \overline{y_0} = 1$, o que mostra que $\overline{y_0}$ (que pertence a $\mathbb{Z}_m^*$) é o inverso de $\overline{a}$. □

Os resultados do Teorema 4 e 5, permitem concluir que $(\mathbb{Z}_m^*, \cdot)$ é um grupo se, e somente se, m é primo.

Em Sistemas de Verificação de Erros, existe um especial interesse em grupos finitos $(G, *)$, tal que o conjunto G é finito. Visto que, os elementos do grupo são os símbolos possíveis numa comunicação, que são naturalmente finitos, do qual será explorado mais a fundo no próximo capítulo. O número de elementos de G é chamado *ordem* do grupo (notação $\mid G \mid$).

Para um conjunto finito a operação pode ser representada em uma tabela, chamada de tábua de operação. Os elementos do conjunto são dispostos como: a primeira fila vertical, sendo a coluna fundamental, e a primeira fila horizontal, sendo linha fundamental.

Exemplo 9. Considere o conjunto $G = \{-1, +1\}$ com a operação de multiplicação usual:

·	1	-1
1		
-1		

Dos quais os resultados das operações é calculado pela operação de cada elemento da coluna fundamental com cada elemento da linha fundamental, ou seja, $1 \cdot 1 = 1, 1 \cdot (-1) = (-1), (-1) \cdot 1 = (-1), (-1) \cdot (-1) = 1$ e representado na tábua de operações da seguinte maneira:

·	1	-1
1	1	-1
-1	-1	1

$G = \{-1, +1\}$ é grupo com a multiplicação usual nos reais, pois G respeita todos os axiomas de grupo, e sua ordem obviamente é 2.

2.2.1 GRUPO DE PERMUTAÇÃO

Permutação é o termo usado na Teoria de Grupos para designar uma bijeção de um conjunto nele mesmo. Um caso particular é $S_3 = \{f_0, f_1, f_2, g_1, g_2, g_3\}$, grupo de Permutação de um conjunto com três elementos:

$$f_0 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \qquad f_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \qquad f_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

$$g_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \qquad g_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} \qquad g_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

Observamos como se obtém, $f_1 \circ g_3$ e $f_2 \circ g_2$ por exemplo:

$$f_1 \circ g_3 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = g_2$$

$$f_2 \circ g_2 = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix} \circ \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix} = g_3$$

De maneira análoga se obtém as demais operações. Feito isso e colocando-se essas operações numa tábua, o resultado será o seguinte:

$\circ$	f_0	f_1	f_2	g_1	g_2	g_3
f_0	f_0	f_1	f_2	g_1	g_2	g_3
f_1	f_1	f_2	f_0	g_3	g_1	g_2
f_2	f_2	f_0	f_1	g_2	g_3	g_1
g_1	g_1	g_2	g_3	f_0	f_1	f_2
g_2	g_2	g_3	g_1	f_2	f_0	f_1
g_3	g_3	g_1	g_2	f_1	f_2	f_0

O conjunto S_3 com a operação $\circ$, é de fato um grupo $(S_3, \circ)$.

Um caso particular importante de grupo de permutações, relacionado com a origem da Teoria de Grupos, é aquele em que $E = \{1, 2, \cdots, n\}$, em que $n \geq 1$, o grupo simétrico de grau n $(S_n, \circ)$. Utilizando a análise combinatória, temos que esse grupo tem ordem $n!$ números de permutações que se podem construir com n elementos.

2.2.2 GRUPO DIEDRAL

Denotando-se por R_0, R_1 e R_2 as rotações em torno de O no sentido anti-horário e por X, Y e Z, respectivamente, as reflexões especiais de π radianos em torno das retas x, y e z, prova-se que o conjunto das simetrias do triângulo é exatamente $\{R_0, R_1, R_2, X, Y, Z\}$. Neste conjunto vamos inserir a operação de composição. Como por exemplo:

Efetuando-se todas as composições possíveis, obtém-se a seguinte tábua:

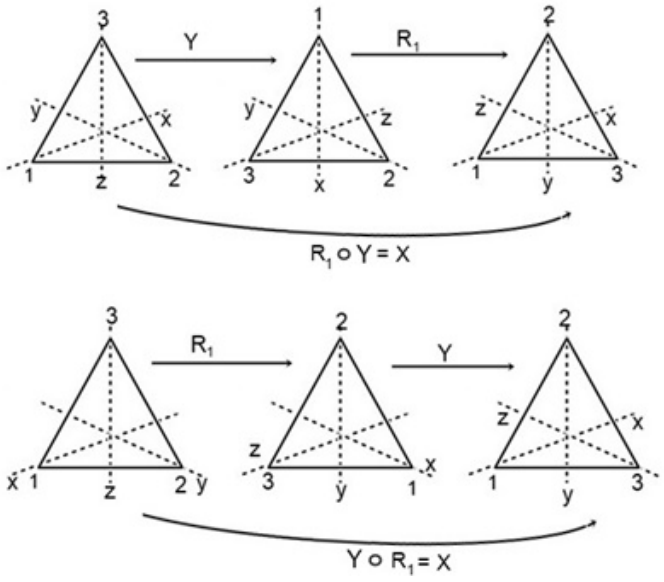

Figura 1: Rotações e Reflexões de um triângulo

$\circ$	R_0	R_1	R_2	X	Y	Z
R_0	R_0	R_1	R_2	X	Y	Z
R_1	R_1	R_2	R_0	Z	X	Y
R_2	R_2	R_0	R_1	Y	Z	X
X	X	Z	Y	R_0	R_2	R_1
Y	Y	X	Z	R_1	R_0	R_2
Z	Z	Y	X	R_2	R_1	R_0

Sendo assim, $(D_3, \circ) = \{R_0, R_1, R_2, X, Y, Z\}$ é grupo.

O conceito de simetria tal como nos casos particulares, o número das simetrias de um polígono regular de n lados é o dobro do número de lados, portanto $2n$ no caso geral.

Isso posto, pode-se demonstrar que o conjunto das simetrias do polígono é

$$D_n = \{R^0, R, R^2, \cdots, R^{n-1}, X, X \circ R, X \circ R^2, \cdots, X \circ R^{n-1}\}.$$

Prosseguindo da mesma forma obtem-se o grupo de simetrias de um pentágono regular $(D_5, \circ) = \{R_0, R_1, R_2, R_3, R_4, X, Y, Z, W, T\}$ um grupo, em que sua tábua de operações é mostrada logo a seguir:

$\circ$	R_0	R_1	R_2	R_3	R_4	X	Y	Z	W	T
R_0	R_0	R_1	R_2	R_3	R_4	X	Y	Z	W	T
R_1	R_1	R_2	R_3	R_4	R_0	Y	Z	W	T	X
R_2	R_2	R_3	R_4	R_0	R_1	Z	W	T	X	Y
R_3	R_3	R_4	R_0	R_1	R_2	W	T	X	Y	Z
R_4	R_4	R_0	R_1	R_2	R_3	T	X	Y	Z	W
X	X	T	W	Z	Y	R_0	R_4	R_3	R_2	R_1
Y	Y	X	T	W	Z	R_1	R_0	R_4	R_3	R_2
Z	Z	Y	X	T	W	R_2	R_1	R_0	R_4	R_3
W	W	Z	Y	X	T	R_3	R_2	R_1	R_0	R_4
T	T	W	Z	Y	X	R_4	R_3	R_2	R_1	R_0

2.3 LOOP

Nesta seção apresentamos algumas noções e resultados preliminares sobre Loops. Segundo (ABC, 2017) a Teoria de Loops originou-se no trabalhos de Ruth Moufang (1905-1977), resultando da integração de conhecimentos e técnicas de álgebra, geometria, topologia e combinatória. Tornou-se uma área de pesquisa muito ativa na Matemática, por ser uma estrutura não associativa.

Definição 7. Um conjunto não vazio L é dito um *loop*, se em L é definido uma operação binária denotado por " $\bullet$ " tal que:

(i) Para todo $a,b \in L$ temos $a \bullet b \in L$ (propriedade de fechamento).

(ii) Existe um elemento $e \in L$ tal que $a \bullet e = e \bullet a = a$, para todo $a \in L$ (e é chamado de elemento neutro de L).

(iii) Para cada par ordenado $(a,b) \in L \times L$ existe um único par $(x,y) \in L \times L$, tal que $ax = b$ e $ya = b$.

No decorrer desse trabalho, vamos considerar somente loops finitos. A operação binária " $\bullet$ " em geral não precisa ser associativa. Todos os grupos são loops, mas nem todo loop é um grupo. Assim, os loops são o conceito mais generalizado de grupos.

Exemplo 10. Considere $L = \{e,a,b,c,d\}$ um loop, com a seguinte composição representado na tábua de operação.

$\bullet$	e	a	b	c	d
e	e	a	b	c	d
a	a	e	c	d	b
b	b	d	a	e	c
c	c	b	d	a	e
d	d	c	e	b	a

Note que, $(b \bullet c) \bullet d = e \bullet d = d \neq b = b \bullet e = b \bullet (c \bullet d)$.

Portanto L é não-associativo.

Definição 8. Um loop $(L, \bullet)$ é dito um *loop comutativo*, se para todo $a, b \in L$ temos que $a \bullet b = b \bullet a$.

Se em um loop $(L, \bullet)$ temos pelo menos um par $a, b \in L$ tal que $a \bullet b \neq b \bullet a$, então dizemos que $(L, \bullet)$ é um loop não-comutativo. O loop do exemplo 10 não é comutativo.

Destacamos a seguinte propriedade de loop, que iremos utilizar no código no capítulo 3.

Proposição 7. Seja $(L, \bullet)$ loop e $a_i, b \in L$ com $i = 1, \cdots, n$. Se $b \neq a_k$, então

$$(a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (a_k \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))))))) \neq (a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (b \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))))))).$$

Demonstração. De fato, tome $r = 0$, e ainda $t = (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))$, logo

$$(a_k \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))) = (a_k \bullet t).$$

Note que $(a_k \bullet t)$ tem solução única pela Definição de Loop. Por hipótese $b \neq a_k$, logo $a_k \bullet t \neq b \bullet t$. Portanto,

$$(a_k \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))) \neq (b \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))).$$

Por outro lado, suponha que seja válido para algum r, iremos provar que vale para $r+1$. Assim,

$$(a_{k+r+1} \bullet \overbrace{(a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (a_k \bullet t))))}^{H.I.}) \neq (a_{k+r+1} \bullet \overbrace{(a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (b \bullet t))))}^{H.I}).$$

Portanto,

$(a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (a_k \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))))))) \neq$

$(a_{k+r} \bullet (\cdots \bullet (a_{k+1} \bullet (b \bullet (a_{k-1} \bullet (\cdots \bullet (a_2 \bullet a_1)))))))).$

□

Nosso intuito em especial, é determinar um código de loop, e para isso, é necessário um loop que tenha exatamente 10 elementos, sendo eles determinados por $0,1,2,3,4,5,6,7,8,9$ que, quando aplicado ao código possa detectar alguns tipos de erros, como veremos no próximo capítulo.

Vejamos um exemplo de um loop com 10 elementos, denominado por $(L_{10}, \bullet)$:

$\bullet$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	0	8	6	3	2	9	7	4
2	2	7	6	1	0	9	5	3	4	8
3	3	9	4	5	7	6	8	1	0	2
4	4	2	3	7	1	0	9	8	5	6
5	5	4	8	9	3	7	0	6	2	1
6	6	8	7	0	2	4	1	5	9	3
7	7	0	9	6	8	2	3	4	1	5
8	8	6	5	2	9	1	4	0	3	7
9	9	3	1	4	5	8	7	2	6	0

Note que $(L_{10}, \bullet)$ não é associativo. De fato, $(1 \bullet 3) \bullet 9 = 8 \bullet 9 = 7 \neq 0 = 1 \bullet 2 = 1 \bullet (3 \bullet 9)$.

3 CÓDIGOS DE VERIFICAÇÃO

3.1 TEORIA DE INFORMAÇÃO

Teoria Matemática da Comunicação ou Teoria da Informação originou-se no pós-guerra com o aperfeiçoamento das máquinas de comunicação desenvolvidas em função da guerra, dando origem à noção de informação como *símbolo calculável* atuando no âmbito da matemática e da engenharia elétrica, e ao nível das telecomunicações. Tendo como objetivo estudar os sistemas de comunicação, tais como transmissão de dados, criptografia, codificação, teoria de ruído, verificação e correção de erros. (INCOMUNIQ, 2011).

A comunicação é baseada em um sistema linear, no qual é entendida como processo de transmissão de uma mensagem por uma fonte de informação, através de um canal a um destinatário, tendo como objetivo transportar a mensagem e ser compreendida tanto por quem gera a mensagem quanto por quem a interpreta.

Um caso particular em que ilustra um sistema de comunicação é uma conversa entre duas pessoas, uma que fala (concepção) e outra que escuta (concessão), a língua falada corresponde ao código da mensagem enquanto o ar desempenha o papel de meio pelo qual as mensagens (ondas sonoras) trafegam, denominada por canal. No entanto, a mensagem enviada pode diferir da mensagem recebida causada por alguma interferência, como um barulho. A informação consiste de ideias que deram origem a mensagem e que se deseja que alcance o receptor/destinatário.

Claude Elwood Shannon (1932-1936) matemático, engenheiro eletrônico e criptógrafo conhecido como *o pai da Teoria da Informação* em 1949, em coautoria com o matemático Warren Weaver (1894-1978), publicou o livro Teoria Matemática da Comunicação (The Mathematical Theory of Communication) (VIEIRA, 2015). Em conjunto desenvolveram um diagrama que representa um sistema geral de comunicação, como mostra na figura 2. Este pode ser adaptado a qualquer nível de comunicação, independentemente das características dos seus componentes (Telefone, TV, rádio, computadores, etc).

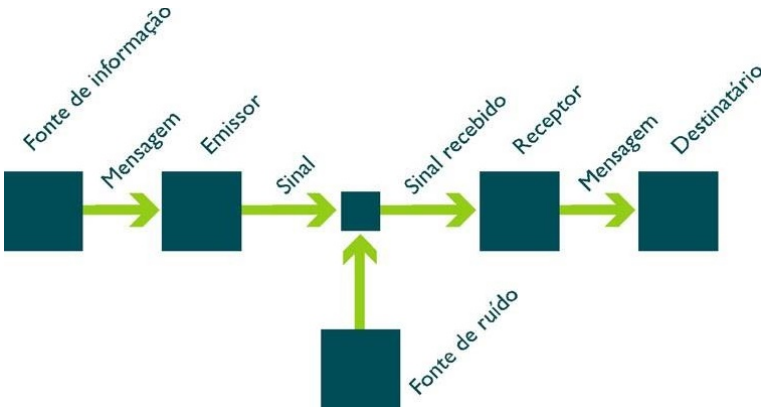

Figura 2: Diagrama de Shannon (INNOVATRIX, 2017)

O livro Teoria Matemática da Comunicação, contém aspectos relacionados com a perda de informação na compressão e na transmissão de mensagens com ruído no canal, além de mostrar que a informação pode ser estudada independente de aspecto semântico e ainda prova que existem códigos que permitem a certificação de que uma informação se transmita com sucesso, mesmo em canais com interferências.

Os erros ocasionados por ruído podem fazer com que a mensagem enviada seja diferente da mensagem recebida, impossibilitando que o destinatário interprete a ideia original. Como no caso particular supracitado, o ruído pode ser entendido de várias formas, como por exemplo: interferência eletromagnética ou sonoras, erros nos equipamentos, erros de digitação no lançamento de informação. Sendo este último tipo de ruído o objetivo deste trabalho.

Para que uma informação seja enviada de forma satisfatória é possível identificar dois problemas principais: as falhas ocorridas nas transmissões dos dados e a segurança destas transmissões. Quanto às falhas de transmissão, para evitar que uma mensagem seja corrompida, Shannon propôs a utilização de redundância (dígito verificador) no código gerador da mensagem para que os erros de transmissão possam ser identificados no destino. Este processo de identificação é chamado de Sistema de Verificação de Erros. Pode-se também criar formas de corrigir erros, dentro de certos parâmetros, identificadas mais a fundo em particular no trabalho de (MILIES, 2009), tal teoria recebe o nome de Códigos Corretores de Erros. Já a Criptografia estuda métodos para modificar as informações, com o objetivo de torná-las acessíveis apenas ao destinatário e que em caso de roubo de informação, ela não seja compreendida.

Frequentemente utilizam-se números de identificação, como por exemplo em cartão de crédito, CNPJ, CPF, matrícula de servidores, título de eleitor, códigos de barras e em várias outras situações. Estes números de identificação são, em geral, formados de algarismos (códigos numéricos) ou de letras e algarismos (códigos alfanuméricos). O uso de números de identificação

possui algumas vantagens, como poder concentrar grande quantidade de informações, e ser entendido em qualquer idioma. No entanto, esses números de identificação não estão isentos dos erros ocasionados pelo ruido, deste modo para detectar a presença de erros utiliza-se a adição de mais informações redundantes chamados de dígitos verificadores, que são obtidos através de operações matemáticas.

3.2 CÓDIGOS VERIFICADORES DE ERROS

Para as próximas seções foram utilizados os trabalhos de (COSTA e PINZ) além dos já mencionados anteriormente.

Um conjunto finito de símbolos A será chamado de alfabeto. Em especial, nos exemplos deste trabalho, temos que $A = \{0, 1, \cdots, 9\}$. Temos ainda que o conjunto com todas as concatenações de n símbolos de A, denotado por $A^n := \{a_1 a_2 \cdots a_n, \ \forall \ \ a_i \in A\}$, é o espaço no qual a informação será transcrita. Os elementos de A^n sao chamados de palavra.

Um Sistema de Verificação de Erros pode ser entendido como uma função que associa cada palavra de A^n com uma palavra de A^{n+r}, $r \in \mathbb{N}^*$. As entradas adicionais as palavras originais referentes a r, são os dígitos verificadores. De forma mais geral os dígitos não precisam ser adicionados no fim da palavra original, mas é o caso de todos os códigos apresentados neste trabalho. Se entende como código todo o conjunto A^{n+r}. Por simplicidade adotaremos no decorrer do trabalho outro tipo de notação.

O dígito verificador não é um número aleatório e o seu valor depende dos demais dígitos da mensagem. Assim, um sistema de verificação de erro faz com que a mensagem acrescida do dígito verificador seja capaz de detectar alguns erros, quando aplicados um código de verificação. Se alguns tipos de erros são cometidos, o dígito verificador será diferente e o erro poderá ser detectado. Apresentamos alguns exemplos de códigos a seguir.

3.2.1 CÓDIGO SOBRE $\mathbb{Z}_M$

Um código de verificação de erros por congruência (mod M) com um dígito verificador, é uma palavra $(a_1, a_2, \cdots, a_{n-1})$, onde deseja-se adicionar o dígito de verificação denotado por a_n. Denotaremos está sequência por um vetor

$$\beta = (a_1, a_2, \cdots, a_{n-1}, a_n).$$

Determina-se um vetor fixo $\omega \in A^n$, chamado vetor de peso, em que é calculado o

produto escalar desses vetores. O dígito de verificação a_n se escolhe de forma tal que a seguinte condição seja respeitada:

$$\beta \cdot \omega \equiv 0 \pmod{M}.$$

A verificação de erro, ocorre quando, por algum erro de digitação, β é alterado para um β' e assim:

$$\beta' \cdot \omega \neq 0 \pmod{M}.$$

No entanto, nem sempre que é cometido erro de digitação em que β é alterado ocorre que $\beta' \cdot \omega \neq 0 \pmod{M}$, pois quando calculado o produto escalar dos vetores poderiam se compensar mutuamente e a soma poderia ainda continuar sendo múltiplo de M.

É possível observar que se tomamos o número M de modo que seja primo e o conjunto β formado por inteiros menores do que M, como cada componente ω_i do vetor de peso é primo relativo com M, resulta que multiplicar por ω_i, em módulo M, equivale a definir uma permutação do conjunto A.

Um código de verificação pode ter mais de um dígito verificador, sendo cada dígito obtido após o outro. Um exemplo em que é usado o código de verificação de erro por congruência com dois dígitos de verificação é o do CPF, como é mostrado a seguir.

CPF

O Cadastro de Pessoas Física (CPF), é o registro de um cidadão na Receita Federal brasileira no qual devem estar todos os contribuintes (pessoas físicas brasileiras ou estrangeiras com negócios no Brasil). O CPF armazena informações fornecidas pelo próprio contribuinte e por outros sistemas da Receita Federal. Sua posse não é obrigatória, mas é necessária em várias situações, como abertura de contas em bancos e emissão de passaporte, por exemplo.

O número de um CPF tem nove dígitos de identificação e mais dois dígitos verificadores que são indicados por último.

O dígito anterior aos dígitos verificadores (isto é, o terceiro dígito da direita para a esquerda) identifica a unidade federativa em que a pessoa se registrou pela primeira vez. Por exemplo, a origem do CPF 043.658.30**6**-27 é Minas Gerais, cujo código é "6". Segue a lista com o número que identifica cada um dos estados brasileiros:

0. Rio Grande do Sul.

1. Distrito Federal, Goias, Mato Grosso, Mato Grosso do Sul e Tocantins.

2. Amazonas, Pará, Roraima, Amapá, Acre e Rondônia.

3. Ceará, Maranhão e Piauí.

4. Paraíba, Pernambuco, Alagoas e Rio Grande do Norte.

5. Bahia e Sergipe.

6. Minas Gerais.

7. Rio de Janeiro e Espírito Santo.

8. São Paulo.

9. Parana e Santa Catarina.

Seja $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}x_{11}$ um número de CPF, onde x_i representa um dígito de identificação para $1 \leq i \leq 9$ e x_{10} e x_{11} são os dígitos verificadores. O algoritmo abaixo, adaptado de (SOUZA, 2013), permite calcular estes dígitos de controle.

$$x_{10} = \left(\sum_{i=1}^{9} ix_i \pmod{11} \right) \pmod{10}$$

$$x_{11} = \left(\sum_{i=2}^{10} (i-1)x_i \pmod{11} \right) \pmod{10}$$

Com a finalidade de ilustrar a aplicação deste algoritmo, vamos verificar a autenticidade do CPF 063.429.523 - 37 calculando os dígitos de controle, x_{10} e x_{11}. Fazendo as devidas substituições obtém-se a seguinte expressão para x_{10}:

$$x_{10} = ((1.0+2.6+3.3+4.4+5.2+6.9+7.5+8.2+9.3) \pmod{11}) \pmod{10}$$

$$x_{10} = ((0+12+9+16+10+54+35+16+27) \pmod{11}) \pmod{10}$$

$$x_{10} = ((179 \pmod{11})) \pmod{10}$$

$$x_{10} = 3 \pmod{10}$$

$$x_{10} = 3$$

Esse resultado confirma o valor do primeiro dígito verificador, agora calcula-se o segundo dígito, x_{11}:

$$x_{11} = ((1.6+2.3+3.4+4.2+5.9+6.5+7.2+8.3+9.3) \pmod{11}) \pmod{10}$$

$$x_{11} = ((6+6+12+8+45+30+14+24+27) \pmod{11}) \pmod{10}$$

$$x_{11} = ((174 \pmod{11})) \pmod{10}$$

$$x_{11} = 7 \pmod{10}$$

$$x_{11} = 7$$

Os cálculos confirmam o valor do segundo dígito de controle. Assim, conclui-se que o CPF 063.429.523 - 37 é autêntico.

É importante ressaltar que o fato de um número de CPF ser autenticado pelos seus dígitos verificadores não o torna um CPF existente. Para isso, é necessário que ele esteja cadastrado no banco de dados da Receita Federal. Assim, um número correto de CPF nem sempre será um documento já emitido. É o que acontece, por exemplo, com números de CPF que têm todos os dígitos iguais, apesar de serem autenticados pelos seus dígitos verificadores, eles não são válidos.

3.2.2 CÓDIGO SOBRE GRUPOS

No Capítulo anterior apresentamos conceitos de grupos de permutações $(S_n, \circ)$, onde (S_n) é um grupo para qualquer conjunto não vazio das permutações de n.

Considere um elemento $c \in A$ qualquer de um grupo $(A, *)$ e n permutações $\sigma^1, \cdots, \sigma^n$ de A, onde cada σ^i é a potência i-ésima de uma permutação σ. O código de verificação de dígitos $C = C(A, \sigma^1, \cdots, \sigma^n, c)$ de comprimento n associado às escolhas $A, \sigma^1, \cdots, \sigma^n$ é o subconjunto de A^n definido por:

$$C := \{(a_1, \cdots, a_n) \in A^n \mid \sigma^1(a_1) * \sigma^2(a_2) * \cdots * \sigma^n(a_n) = c\}.$$

Como qualquer palavra $a_1 \cdots a_n \in C$ satisfaz $\sigma^1(a_1) * \sigma^2(a_2) * \cdots * \sigma^n(a_n) = c$.

Em 1969 Verhoeff, na sua tese de doutorado, desenvolveu um método simples, baseado não em cálculos com números inteiros, mas com os elementos de um certo grupo (D_5). Procedendo dos conceitos apresentados no Capítulo 2 temos a seguinte tábua de operação do $(D_5, \cdot)$:

·	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Note que as funções de D_5 da página 19, foram renomeadas para a facilidade na manipulação da mesma, de modo que $0,1,2,3,4$ para designar as rotações correspondentes e $5,6,7,8,9$ para as respectivas reflexões.

Considere ainda a seguinte permutação:

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 5 & 7 & 6 & 2 & 8 & 3 & 0 & 9 & 4 \end{pmatrix}.$$

A proposta de Verhoeff consiste em transformar um vetor de informação $(a_1, \cdots, a_{n-1})$ num vetor de codificação, adicionando um dígito de verificação a_n de forma tal que

$$\sigma(a_1) * \sigma^2(a_2) * \cdots * \sigma^{n-1}(a_{n-1}) * a_n = 0 \text{ em } D_5. \tag{1}$$

Suponha um erro na posição k, assim obtem-se tal expressão:

$$\sigma(a_1) * \cdots * \sigma^{k-1}(a_{k-1}) * \sigma^k(b) * \sigma^{k+1}(a_{k+1}) * \cdots * \sigma^{n-1}(a_{n-1}) * a_n.$$

Usando a associatividade temos:

$$(\sigma(a_1) * \cdots * \sigma^{k-1}(a_{k-1})) * \sigma^k(b) * (\sigma^{k+1}(a_{k+1}) * \cdots * \sigma^{n-1}(a_{n-1}) * a_n) \neq 0.$$

Pois, tanto quanto o lado esquerdo da posição do erro, quanto o direito são iguais a equação 1 e pela solubilidade das equações em grupos garante a detecção de um erro na posição k qualquer.

O sistema de Verhoeff também verifica erro de transposição adjacente. Isso se deve a

uma propriedade de σ em relação a D_5, chamada de antissimétrica.

Definição 9. Uma permutação σ de um grupo $(G,*)$ diz-se uma aplicação antissimétrica se verifica a seguinte condição:

$$x*\sigma(y) \neq y*\sigma(x), \text{ para todo par de elementos } x,y \in G.$$

A permutação σ de Verhoeff é antissimétrica (MILIES; MILIES, 2013,p.17).

Suponha um erro de tranposição adjacente da forma:

$$(\sigma(a_1)*\cdots*[\sigma^k(a_{k+1})*\sigma^{k+1}(a_k)]*\cdots*\sigma^{n-1}(a_{n-1})*a_n) \neq 0.$$

Considerando a definição 9, tomando $x=\sigma^k(a_k)$ e $y=\sigma^k(a_{k+1})$, temos:

$$\begin{aligned}\sigma^k(a_k)*\sigma(\sigma^k(a_{k+1})) &= \sigma^k(a_k)*\sigma^{k+1}(a_{k+1})\\ &\neq \sigma^k(a_{k+1})*\sigma^{k+1}(a_k)\\ &= \sigma^k(a_{k+1})*\sigma(\sigma^k(a_k)).\end{aligned}$$

Ou seja, todo erro de transposição adjacente, quando aplicado a uma permutação antissimétrica será detectado.

Um exemplo em que é usado o código de verificação de erro dado pelo grupo D_5 é o do Marco Alemão como é mostrado seguir.

Marco Alemão

Outro exemplo interessante, embora não cotidiano, é a numeração utilizada pelo Deusche Bundesbank, órgão emissor de dinheiro da Alemanha, utilizado antes do Euro.

Figura 3: Cédula de Marco Alemão (PINZ, 2013)

As cédulas do Marco Alemão são identificadas com 11 dígitos. Além dos dez algarismos utiliza as letras A, D, G, K, L, N, S, U, V e Z. Para verificar a validade da nota, as letras

são trocadas por números, conforme a tabela:

A	D	G	K	L	N	S	U	V	Z
0	1	2	3	4	5	6	7	8	9

O código usado pelo banco, para a verificação da validade da cédula, utiliza a tabela de operação do grupo D_5, ao invés de utilizar uma permutação e suas potências, utiliza dez permutações diferentes.

	0	1	2	3	4	5	6	7	8	9
σ^1	1	5	7	6	2	8	3	0	9	4
σ^2	5	8	0	3	7	9	6	1	4	2
σ^3	8	9	1	6	0	4	3	5	2	7
σ^4	9	4	5	3	1	2	6	8	7	0
σ^5	4	2	8	6	5	7	3	9	0	1
σ^6	2	7	9	3	8	0	6	4	1	5
σ^7	7	0	4	6	9	1	3	2	5	8
σ^8	0	1	2	3	4	5	6	7	8	9
σ^9	1	5	7	6	2	8	3	0	9	4
σ^{10}	5	8	0	3	7	9	6	1	4	2

Vamos verificar a validade da cédula da figura 3, o número de série da cédula é $AA3457494N2$. Utilizando a tabela, pode-se reescrever a numeração somente com algarismos 00345749452.

Aplicamos ordenadamente as permutações dadas na tabela anterior

Por $\sigma^1 : 0 \to 1$, $\sigma^2 : 0 \to 5, \sigma^3 : 3 \to 6, \sigma^4 : 4 \to 1, \sigma^5 : 5 \to 7, \sigma^6 : 7 \to 4, \sigma^7 : 4 \to 9, \sigma^8 : 9 \to 9, \sigma^9 : 4 \to 2, \sigma^{10} : 5 \to 9$ e a última é fixa, $2 \to 2$.

Agora operaremos com os resultados das permutações. Para isso, utilizamos a tábua de operações de D_5:

$$\sigma^1(0) * \sigma^2(0) = 1 * 5 = 6 \quad \to \quad 6 * \sigma^3(3) = 6 * 6 = 0 \quad \to \quad 0 * \sigma^4(4) = 0 * 1 = 1 \quad \to$$

$$1 * \sigma^5(5) = 1 * 7 = 8 \quad \to \quad 8 * \sigma^6(7) = 8 * 4 = 9 \quad \to \quad 9 * \sigma^7(4) = 9 * 9 = 0 \quad \to$$

$$0 * \sigma^8(9) = 0 * 9 = 9 \quad \to \quad 9 * \sigma^9(4) = 9 * 2 = 7 \quad \to \quad 7 * \sigma^{10}(5) = 7 * 9 = 3 \quad \to \quad 3 * 2 = 0.$$

De onde podemos concluir que a cédula tem numeração válida. Abordando estes exemplos, no próximo capítulo, destacamos eficácia dos códigos conforme Verhoeff.

3.3 TABELA DE VERHOEFF E ANÁLISE DE ALGUNS CÓDIGOS

Jacobus Koos Verhoeff, nasceu em 1927 na Holanda, estudou matemática em Leiden e Amsterdã, teve uma carreira muito variada, trabalhou no centro de pesquisa de matemática em Amsterdã, conhecido como Centro Matemático (agora Centro de Matemática e Ciência da Computação) e ensinava na Universidade Tecnológica de Delft. Mais tarde, ele trabalhou na indústria da Philips em Eindhoven. Finalmente, tornou-se professor titular de Informática na Universidade Erasmus de Roterdã (AMS, 2017).

Verhoeff escreveu sua dissertação de doutorado na área da Teoria da Codificação. O matemático holandês foi responsável pelo sistema de códigos decimais baseado no grupo diédrico da ordem 10 (D_5) publicado em *Error Detecting Codes*. Entretanto, a publicação em que o seu sistema de dígitos de verificação está incluído foi uma versão retrabalhada de sua tese.

Os erros cometidos ao digitar um número foram sistematicamente investigados por autores como Beckley e Verhoeff. Verhoeff concluiu sua pesquisa considerando que 79% dos erros ocorrem com a digitação equivocada de um único dígito, como, por exemplo: digitar 3872, em que o correto seria 3**9**72, quando isto acontece este tipo de erro recebe o nome de erro singular.

Os erros de transposição representam 11% em que se dividem em dois casos: adjacentes com 10,2% e alternada com 0,8%. O primeiro tipo refere-se à troca de posição de dois dígitos diferentes situados lado a lado, como por exemplo: digitar 3792, em que o correto seria 3**97** 2. O segundo, refere-se à troca de posição de dois dígitos diferentes alternados por um terceiro dígito. Por exemplo: digitar 3**2**7**9**.

Os demais 10% dos erros estão distribuídos em diversas categorias; erro gêmeo, erro gêmeo alternado, e outros. Estes estudos também nos dizem que a incidência de mais de um erro ao digitar um número é muito pouco provável.

Tipo de erro	Frequência relativa
erro único $...a... \mapsto ...b...$	79%
transposição adjacente $...ab... \mapsto ...ba...$	$10,2\%$
transposição alterna $...abc... \mapsto ...cba...$	$0,8\%$
erro gêmeo $...aa... \mapsto ...bb...$	$0,6\%$
erro gêmeo alternado $...aba... \mapsto ...cbc...$	$0,3\%$
outros	$9,1\%$

Figura 4: Tipos de erros e suas frequências segundo Verhoeff (PINZ, 2013)

A Teoria de Sistemas Verificadores não quer somente analisar os possíveis erros, mas sim detectar os erros mais comuns. Considere o sistema de verificação do CPF, discutido anteriormente. Imaginemos que seja criado um sistema para detectar o erro de permutar os dígitos a_3 e a_8. A construção matemática de tal sistema pode vir a ser interessante, mas a chance de tal erro ser cometido é bastante baixa.

Quando é utilizado o módulo 11 para o cálculo do dígito verificador, há uma particularidade: o resto da divisão por 11 pode ser 10. Isso exige uma forma especial para representar este resto. Para se usar apenas um dígito verificador, são adotadas duas soluções diferentes: utilizar o caractere X para representar o resto 10 chamado de módulo 11 completo (utilizado na numeração das agências de alguns bancos no Brasil, por exemplo), e outra possibilidade utiliza o dígito 0 para representar o resto 10 chamado de módulo 11 restrito. Note que no caso do CPF é utilizado ((módulo 11) módulo 10) para que o problema do resto 10 seja suprido, visto que é equivalente a utilizar o módulo 11 restrito.

O uso de dois dígitos verificadores no CPF diminui os casos de falha de detecção em relação a usar apenas um dígitos com o módulo 11 restrito.

No exemplo do Marco Alemão, se acontecer uma transposição adjacente entre os dígitos a_7 e a_8, ou seja, o número de série da cédula é trocada por 003457**94**452, e a verificação acontece da seguinte forma:

Por σ^1 temos $0 \to 1$, $\sigma^2 : 0 \to 5, \sigma^3 : 3 \to 6, \sigma^4 : 4 \to 1, \sigma^5 : 5 \to 7, \sigma^6 : 7 \to 4, \sigma^7 : 9 \to 8, \sigma^8 : 4 \to 4, \sigma^9 : 4 \to 2, \sigma^{10} : 5 \to 9$ e a última é fixa, $2 \to 2$.

Agora operaremos com os resultados das permutações. Para isso, utilizamos a tábua da operação composição D_5, temos:

$$\sigma^1(0) * \sigma^2(0) = 1 * 5 = 6 \quad \to \quad 6 * \sigma^3(3) = 6 * 6 = 0 \quad \to \quad 0 * \sigma^4(4) = 0 * 1 = 1 \quad \to$$

$$1 * \sigma^5(5) = 1 * 7 = 8 \quad \to \quad 8 * \sigma^6(7) = 8 * 4 = 9 \quad \to \quad 9 * \sigma^7(9) = 9 * 8 = 1 \quad \to$$

$$1 * \sigma^8(4) = 1 * 4 = 0 \quad \to \quad 0 * \sigma^9(4) = 0 * 2 = 2 \quad \to \quad 2 * \sigma^{10}(5) = 2 * 9 = 6 \quad \to \quad 6 * 2 = 9.$$

Como $9 \neq 0$ temos que o erro seria detectado.

Este método, porém, tem um inconveniente. Nos cálculos, ele não distingue entre uma letra e o número que lhe é associado. Assim por exemplo, se a letra K for trocada pelo número 3, o sistema será incapaz de detectar o erro. O mesmo acontece se ocorre uma transposição de 3 e K, ou vice-versa. Para evitar este problema, poder-se-ia usar o grupo D_{18}, que tem 36 elementos (e portanto os vinte símbolos usados no código alfanumérico das notas corresponderiam a

elementos diferentes em D_5), com uma permutação adequada.

O método apresentado por Verhoeff, com os componentes do grupo diedral D_5 que também detecta todos os erros únicos e todas as transposições adjacentes, sem a necessidade de símbolos extras. Ou seja, o Sistema de Verificação de Erro presente no Marco Alemão detecta 89,2%.

3.4 CÓDIGO SOBRE LOOPS

Considerando as ideias de Verhoeff na construção do código utilizado no marco alemão, vamos propor um código utilizando um loop, no lugar do grupo D_5.

Considere um elemento $c \in L$ qualquer de um loop $(L, \bullet)$ e n permutações $\sigma^1, \cdots, \sigma^n$ de L. O código de verificação de dígitos $C = C(L, \sigma^1, \cdots, \sigma^n, c)$ de comprimento n associado às escolhas $L, \sigma^1, \cdots, \sigma^n$ é o subconjunto de L^n definido por:

$$C := \{(a_1, \cdots, a_n) \in L^n \mid [\sigma^1(a_1) \bullet [\sigma^2(a_2) \bullet [\cdots \bullet \sigma^n(a_n)]]] = c\}.$$

Como qualquer palavra $a_1 \cdots a_n \in C$ satisfaz $[\sigma^1(a_1) \bullet [\sigma^2(a_2) \bullet [\cdots \bullet \sigma^n(a_n)]]] = c$.

Pode-se transformar um vetor de informação $(a_1, \cdots, a_{n-1})$ em um vetor de codificação, adicionando um dígito de verificação a_n de forma tal que:

$$[\sigma(a_1) \bullet [\sigma^2(a_2) \bullet [\cdots \bullet [\sigma^{n-1}(a_{n-1}) \bullet a_n]]]] = 0 \text{ em } L. \tag{2}$$

Note que, o sistema proposto acima tem a mesma estrutura que o sistema de Verhoeff, dada pela equação 1, havendo somente a mudança do grupo para um loop.

No código utilizado no marco alemão, Verhoeff propõe uma permutação antissimétrica, ou seja $x * \sigma(y) \neq y * \sigma(x)$, que garante a detecção de erros de transposição adjacente nos códigos sobre grupo, conforme discutido em na seção 3.2.2. Vale lembrar que a associatividade é essencial para que esta propriedade seja válida. Como um loop não é associativo, a permutação antissimétrica do grupo não é válido, sendo assim, vamos buscar uma propriedade equivalente para loops.

Proposição 8. O sistema proposto na equação 2 detecta erro único.

Demonstração. Seja $[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_k) \bullet [\sigma^{k+1}(a_{k+1}) \bullet [\cdots \bullet [\sigma^{n-1} \bullet (a_n)]]]]]] = 0$. Se trocar $a_k \to b$, ou seja $[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(b) \bullet [\sigma^{k+1}(a_{k+1}) \bullet [\cdots \bullet [\sigma^{n-1} \bullet (a_n)]]]]]]$, temos que, como σ^k

é uma bijeção, pela injetividade $\sigma^k(a_k) \neq \sigma^k(b)$. Pela proposição 7 segue que o código detecta erro único.

□

Definição 10. Uma permutação σ de um loop L, diz-se uma aplicação ótima se verifica a seguinte condição:

$$x \bullet [\sigma(y) \bullet t] \neq y \bullet [\sigma(x) \bullet t], \qquad \forall\ x,y,t \in L. \tag{3}$$

A próxima proposição garante que dado uma aplicação ótima, todo erro de transposição adjacente será detectado.

Proposição 9. Se σ é uma permutação ótima em relação a um loop L, então o código gerado por σ e L detecta todo erro de transposição adjacente.

Demonstração. Considere $(L,\bullet)$ um loop e $a_i \in L$, com $i = 1,\cdots,n$. Queremos provar que

$$[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_k) \bullet [\sigma^{k+1}(a_{k+1}) \bullet [\cdots \bullet [\sigma^{n-1} \bullet (a_n)]]]]]] \neq$$

$$[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_{k+1}) \bullet [\sigma^{k+1}(a_k) \bullet [\cdots \bullet [\sigma^{n-1} \bullet (a_n)]]]]]].$$

Tome $t = [\sigma^{k+2}(a_{k+2}) \bullet [\cdots \bullet [\sigma^{n-1}(a_{n-1}) \bullet (a_n)]]]$, assim temos

$$[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_k) \bullet [\sigma^{k+1}(a_{k+1}) \bullet t]]]].$$

Considere agora $x = \sigma^k(a_k)$ e $y = \sigma^k(a_k + 1)$

Por hipótese σ é ótima, logo

$$x \bullet [\sigma(y) \bullet t] \neq y \bullet [\sigma(x) \bullet t] \qquad \Leftrightarrow$$

$$\sigma^k(a_k) \bullet [\sigma(\sigma^k(a_k + 1)) \bullet t] \neq \sigma^k(a_k + 1) \bullet [\sigma(\sigma^k(a_k))] \bullet t] \qquad \Leftrightarrow$$

$$\sigma^k(a_k) \bullet [\sigma^{k+1}(a_k + 1) \bullet t] \neq \sigma^k(a_k + 1) \bullet [\sigma^{k+1}(a_k) \bullet t].$$

Pela definição de loop, se $\sigma^k(a_k) \bullet [\sigma^{k+1}(a_k + 1) \bullet t] \neq \sigma^k(a_k + 1) \bullet [\sigma^{k+1}(a_k) \bullet t]$, então

$$[\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_k) \bullet [\sigma^{k+1}(a_{k+1}) \bullet t]]]] \neq [\sigma(a_1) \bullet [\cdots \bullet [\sigma^k(a_{k+1}) \bullet [\sigma^{k+1}(a_k) \bullet t]]]].$$

Portanto o código de loop detecta todo erro de transposição adjacente.

□

Na busca por encontrar um loop com 10 elementos, que tenha uma permutação com aplicação ótima, utilizamos o programa GAP System.

> GAP é um sistema para álgebra discreta computacional, com ênfase em Teoria de Grupos Computacional. Gap provê uma linguagem de programação, um acervo com milhares de funções implementando algoritmos escritos na linguagem Gap, bem como um grande acervo de objetos algébricos.[1] (GAP, 2018, online, tradução nossa)

Utilizando loops gerados aleatoriamente, não conseguimos encontrar um loop de 10 elementos que tenha uma permutação σ que respeita a condição ótima. Assim, para o nosso código de loop, utilizamos uma permutação que tenha uma quantidade mínima de casos em que a equação 3 não é respeitada para um certo loop, dado aleatoriamente pelo GAP. A tábua de tal loop é dada abaixo.

$\bullet$	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	0	8	6	3	2	9	7	4
2	2	7	6	1	0	9	5	3	4	8
3	3	9	4	5	7	6	8	1	0	2
4	4	2	3	7	1	0	9	8	5	6
5	5	4	8	9	3	7	0	6	2	1
6	6	8	7	0	2	4	1	5	9	3
7	7	0	9	6	8	2	3	4	1	5
8	8	6	5	2	9	1	4	0	3	7
9	9	3	1	4	5	8	7	2	6	0

E a permutação é dada por:

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 0 & 7 & 1 & 9 & 2 & 3 & 4 & 8 & 6 & 5 \end{pmatrix}$$

Considerando a permutação e suas potências, obtém-se a seguinte tabela de operação:

[1] GAP is a system for computational discrete algebra, with particular emphasis on Computational Group Theory. GAP provides a programming language, a library of thousands of functions implementing algebraic algorithms written in the GAP language as well as large data libraries of algebraic objects.

	0	1	2	3	4	5	6	7	8	9
σ^1	0	7	1	9	2	3	4	8	6	5
σ^2	0	8	7	5	1	9	2	6	4	3
σ^3	0	6	8	3	7	5	1	4	2	9
σ^4	0	4	6	9	8	3	7	2	1	5
σ^5	0	2	4	5	6	9	8	1	7	3
σ^6	0	1	2	3	4	5	6	7	8	9

Vamos verificar a validade de um código aleatório, com 6 dígitos e 1 dígito de verificação, dado por 4539141. Utilizando a equação 2, temos:

$$[\sigma^1(4) \bullet [\sigma^2(5) \bullet [\sigma^3(3) \bullet [\sigma^4(9) \bullet [\sigma^5(1) \bullet [\sigma^6(4) \bullet 1]]]]]].$$

Aplicando ordenadamente as permutações dadas na tabela anterior:

Por $\sigma^1 : 4 \to 2$, $\sigma^2 : 5 \to 9$, $\sigma^3 : 3 \to 3$, $\sigma^4 : 9 \to 5$, $\sigma^5 : 1 \to 2$, $\sigma^6 : 4 \to 4$, e a última é fixa, $1 \to 1$.

Agora operaremos com os resultados das permutações. Para isso, utilizamos a tábua de operações de L_{10}, temos:

$$\sigma^6(4) \bullet 1 = 4 \bullet 1 = 2 \quad \to \quad \sigma^5(1) \bullet 2 = 2 \bullet 2 = 6 \quad \to \quad \sigma^4(9) \bullet 6 = 5 \bullet 6 = 0 \quad \to$$
$$\sigma^3(3) \bullet 0 = 3 \bullet 0 = 3 \quad \to \quad \sigma^2(5) \bullet 3 = 9 \bullet 3 = 4 \quad \to \quad \sigma^1(4) \bullet 4 = 2 \bullet 4 = 0.$$

De onde podemos concluir que o código é válido. Suponha agora que um erro único é cometido e o código que antes era 4539141, passa a ser 4537141. Utilizando a equação 2, temos:

$$[\sigma^1(4) \bullet [\sigma^2(5) \bullet [\sigma^3(3) \bullet [\sigma^4(7) \bullet [\sigma^5(1) \bullet [\sigma^6(4) \bullet 1]]]]]].$$

Aplicando ordenadamente as permutações dadas na tabela anterior:

Por $\sigma^1 : 4 \to 2$, $\sigma^2 : 5 \to 9$, $\sigma^3 : 3 \to 3$, $\sigma^4 : 7 \to 2$, $\sigma^5 : 1 \to 2$, $\sigma^6 : 4 \to 4$, e a última é fixa, $1 \to 1$.

Agora operaremos com os resultados das permutações. Para isso, utilizamos a tábua de operações de L_{10}, temos:

$$\sigma^6(4) \bullet 1 = 4 \bullet 1 = 2 \quad \to \quad \sigma^5(1) \bullet 2 = 2 \bullet 2 = 6 \quad \to \quad \sigma^4(7) \bullet 6 = 2 \bullet 6 = 5 \quad \to$$
$$\sigma^3(3) \bullet 5 = 3 \bullet 5 = 6 \quad \to \quad \sigma^2(5) \bullet 6 = 9 \bullet 6 = 7 \quad \to \quad \sigma^1(4) \bullet 4 = 2 \bullet 7 = 3.$$

Como $3 \neq 0$ temos que o erro único será detectado. Suponha agora um erro de transposição adjacente, tal como **54**39141, utilizando a equação 2, temos:

$$[\sigma^1(5) \bullet [\sigma^2(4) \bullet [\sigma^3(3) \bullet [\sigma^4(9) \bullet [\sigma^5(1) \bullet [\sigma^6(4) \bullet 1]]]]]].$$

Aplicando ordenadamente as permutações dadas na tabela anterior:

Por $\sigma^1 : 5 \to 3$, $\sigma^2 : 4 \to 1$, $\sigma^3 : 3 \to 3$, $\sigma^4 : 9 \to 5$, $\sigma^5 : 1 \to 2$, $\sigma^6 : 4 \to 4$, e a última é fixa, $1 \to 1$.

Agora operaremos com os resultados das permutações. Para isso, utilizamos a tábua de operações de L_{10}, temos:

$$\sigma^6(4) \bullet 1 = 4 \bullet 1 = 2 \quad \to \quad \sigma^5(1) \bullet 2 = 2 \bullet 2 = 6 \quad \to \quad \sigma^4(9) \bullet 6 = 5 \bullet 6 = 0 \quad \to$$
$$\sigma^3(3) \bullet 0 = 3 \bullet 0 = 3 \quad \to \quad \sigma^2(4) \bullet 3 = 1 \bullet 3 = 8 \quad \to \quad \sigma^1(5) \bullet 8 = 3 \bullet 8 = 0.$$

Donde podemos concluir que o código é valido mesmo com um erro de transposição adjacente, ou seja o erro não foi detectado.

Apesar de o loop proposto não ter uma permutação ótima, garantindo assim detectar todo erro de transposição adjacente, σ desrespeita a equação 3, em 16 casos de "x, y e t", dos 900 possíveis detectando assim 98,2% dos erros de transposição adjacente.

4 CONCLUSÃO

Destacamos a importância das preliminares algébricas, visto que não seria possível construir a teoria de verificação sem os conceitos matemáticos apresentados no referencial teórico. Os conceitos de Teoria de Números, Grupos e Loops são fundamentais, dando suporte e tornando possível realizar as verificações necessárias.

Verhoeff, em seus estudos propôs uma tabela com as frequências relativas aos erros mais comuns, apresentados neste trabalho. Garantir a eficiência de um código, não é simplesmente pela *quantidade* de erros que ele detecta, mas pela *qualidade* desses erros, isso é, um sistema eficiente é aquele que detecta os erros mais prováveis. Por isso a tabela de Verhoeff é considerada importante para o estudo de códigos.

O Código de Loop, proposto nesse trabalho, detecta todo erro único, além de detectar 98,2% dos erros de transposição adjacente. Comparado com o código de Verhoff, esse código se mostra eficiente, porém mais limitado, visto que não detecta todos os erros de transposição adjacente. Vale ressaltar que essa limitação se deve a nossa *incapacidade* de determinar um loop com uma permutação ótima. O método proposto nesse trabalho para determinar tal par (loop e permutação) é bastante primitivo, visto que busca esse par aleatoriamente utilizando o GAP-System. Destacamos o potencial dessa busca como tema para novos projetos de pesquisa, através de metodologias computacionais ou algébricas.

REFERÊNCIAS

ABC, U. F. do. **Brazilian Meeting on Loops and Nonassociative Systems**. 2017. Disponível em: <http://hostel.ufabc.edu.br/maria.giuliani/brazilianmeeting-br.html>. Acesso em: 09 de maio de 2018.

ALEKSANDROV, A. D.; KOLMOGOROV, A. N. **La matemática: su contenido, métodos y significado, 1**. 3. ed. Madrid, España: Alianza Editorial, 1994. 310 p.

AMS. **American Mathematical Society**. 2017. Disponível em: <http://www.ams.org/samplings/feature-column/fcarc-verhoeff>. Acesso em: 17 de junho de 2017.

BOYER, C. B.; MERZBACH, U. C. História da matemática. [rev.]. **São Paulo: Edgard Blücher LTDA**, 1996.

BROWN, D. Biquinary decimal error detection codes with one, two and three check digits. **The Computer Journal**, Br Computer Soc, v. 17, n. 3, p. 201–204, 1974.

COSTA, F. R. A. **Sistemas de Identificaçao Modular: uma aplicaçao no ensino fundamental**. Dissertação (Mestrado), 2014.

DOMINGUES, H. H.; IEZZI, G. **Álgebra moderna**. 3. ed. São Paulo: Atual, 2003.

GAP. **GAP - Groups, Algorithms, Programming - a System for Computational Discrete Algebra**. 2018. Disponível em: <http://www.gap-system.org/>. Acesso em: 21 de maio de 2018.

INCOMUNIQ. **Teoria Matemática da Comunicação**. 2011. Disponível em: <http://incomuniq.blogspot.com/2011/11/teoria-matematica-da-comunicacao>. Acesso em: 17 de maio de 2018.

INNOVATRIX. **O processo de aprender e o diagrama de Shannon**. 2017. Disponível em: <http://innovatrix.com.br/o-diagrama-que-define-a-informacao/>. Acesso em: 17 de junho de 2017.

KANDASAMY, W. V. **Smarandache loops**. 1. ed. Rehoboth, NM: Infinite Study, 2002.

LARSEN, H. L. Generalized double modulus 11 check digit error detection. **BIT Numerical Mathematics**, Springer, v. 23, n. 3, p. 303–307, 1983.

MELO, H. S. Gauss: o príncipe da matemática. **Correio dos Açores**, Gráfica Açoreana, Lda., p. 14–14, 2015.

MILIES, C. P. Breve introduçaoa teoria dos códigos corretores de erros. **Departamento de Matemática, UFMS**, 2009.

MILIES, C. P.; MILIES, C. ou. A matemática dos códigos de barras. IME/USP, 2013.

MILIES, F. C. P.; COELHO, S. P. **Números: uma introdução à matemática**. 3. ed. São Paulo: Edusp, 2001.

PINTO, M. M. P. **Grupos e simetrias.** 2 p. Dissertação (Mestrado), 2009.

PINZ, C. R. F. **Dígitos verificadores e detecção de erros**. Dissertação (Mestrado), 2013.

SÁ, C. C. de; ROCHA, J. **Treze viagens pelo mundo da matemática**. 1. ed. Porto: Universidade do Porto, 2010.

SOUZA, N. P. **Uma análise dos esquemas de dígitos verificadores usados no Brasil**. Dissertação (Mestrado), 2013.

USP. **Joseph-Louis Lagrange**. 2012. Disponível em: <http://ecalculo.if.usp.br/historia/lagrange>. Acesso em: 21 de maio de 2018.

VIEIRA, E. **Biografia de Claude Shannon**. 2015. Disponível em: <http://biografiaecuriosidade.blogspot.com/2015/11/biografia-de-claude-shannon>. Acesso em: 21 de maio de 2018.