Uma abordagem de dígitos verificadores e códigos corretores no Ensino Fundamental

Daniel Alves Machado

Daniel Alves Machado

Uma abordagem de dígitos verificadores e códigos corretores no Ensino Fundamental

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre – Programa de Mestrado Profissional em Matemática. *VERSÃO REVISADA*

Área de Concentração: Matemática

Orientador: Prof. Dr. Marcelo Rempel Ebert

USP – São Carlos
Junho de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

M149u

Machado, Daniel Alves
Uma abordagem de dígitos verificadores e códigos corretores no Ensino Fundamental / Daniel Alves Machado; orientador Marcelo Rempel Ebert. - São Carlos - SP, 2016.
63 p.

Dissertação (Mestrado - Programa de Pós-graduação em Mestrado Profissional em Matemática em Rede Nacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

1. Dígitos verificadores. 2. Códigos corretores. 3. Códigos lineares. 4. Métrica de Hamming. I. Ebert, Marcelo Rempel, orient. II. Título.

Daniel Alves Machado

An approach to check digits and error-correcting codes in middle school

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master – Program in Mathematics Professional Master. *FINAL VERSION*

Concentration Area: Mathematics

Advisor: Prof. Dr. Marcelo Rempel Ebert

USP – São Carlos
June 2016

Dedico este trabalho ao meu afilhado Matheus Alexandre, para que sempre acredite em seus sonhos e nunca deixe de estudar.

AGRADECIMENTOS

Ao meu orientador, professor Marcelo Rempel Ebert, por me orientar na elaboração do trabalho.

A Madelaine Pires, pela ajuda com a correção ortográfica e gramatical.

À minha família e a minha namorada Carolina Binbanco, pelo apoio e paciência.

Aos alunos e à equipe da EMEF Eponina de Brito Rossetto, que possibilitaram a aplicação da proposta pedagógica elaborada.

Aos meus companheiros de turma, pela ajuda mútua no decorrer do curso.

À professora Vanessa Rolnik Artioli, pelo ensinamento de como utilizar o programa *Latex*.

Ao professor Luís Amilo, que sempre me inspirou.

A CAPES pelo apoio financeiro.

"A Matemática apresenta invenções tão sutis
que poderão servir não só para satisfazer os curiosos como,
também para auxiliar as artes e poupar trabalho aos homens."
(Descartes)

RESUMO

MACHADO, D. A.. **Uma abordagem de dígitos verificadores e códigos corretores no Ensino Fundamental**. 2016. 63 f. Dissertação (Mestrado – Programa de Mestrado Profissional em Matemática) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Este trabalho, elaborado por meio de pesquisa bibliográfica, apresenta um apanhado sobre os dígitos verificadores presentes no Cadastro de Pessoas Físicas (CPF), no código de barras, e no sistema ISBN; faz uma introdução sobre a métrica de Hamming e os códigos corretores de erros; cita a classe de códigos mais utilizada, que são os códigos lineares, e deixa a sugestão de uma proposta pedagógica para professores de matemática aplicarem no Ensino Fundamental, podendo ser ajustada também para o Ensino Médio. No apêndice A, são propostos alguns exercícios que podem ser trabalhados com os alunos em sala de aula.

Palavras-chave: Dígitos verificadores, Códigos corretores, Códigos lineares, Métrica de Hamming.

ABSTRACT

MACHADO, D. A.. **Uma abordagem de dígitos verificadores e códigos corretores no Ensino Fundamental**. 2016. 63 f. Dissertação (Mestrado – Programa de Mestrado Profissional em Matemática) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

This work, based on the attached references, presents an overview of the check digits that appear in the Brazilian document CPF, in the bar code and the ISBN system. Moreover, it makes an introduction to the Hamming metric and error-correcting codes. In particular, some considerations about linear codes are done and it makes a suggestion of a pedagogical approach to apply it in middle school and can also be adjusted to high school. In the Appendix A are proposed some exercises to students.

Key-words: Check digits, Error-correcting codes, Linear codes, Hamming metric.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Os códigos estão presentes na vida das pessoas, e nós os utilizamos a todo momento, para pagar uma conta, para comprar um produto, para se comunicar; o nosso próprio alfabeto é um exemplo de código.

Os códigos numéricos estão presentes nos documentos pessoais, códigos postais, boletos bancários, entre outros. Seu uso possui algumas vantagens, como poder registrar uma quantidade maior de informações, e ser entendido em qualquer idioma, contudo, sua principal desvantagem é a dificuldade de detectar a presença de erros.

Para contornar essa desvantagem, alguns códigos acrescentam dígitos verificadores, que são obtidos por meio de operações matemáticas com os demais dígitos do código, alguns exemplos de dígitos verificadores e seus algoritmos de cálculo serão apresentados no Capítulo 2, baseado em (FINI, 2009).

Os dígitos verificadores permitem apenas detectar alguns tipos de erros, mas não oferece formas para corrigi-los, para isso, é preciso acrescentar dígitos de redundância, semelhantes aos dígitos verificadores, porém em maior quantidade. No Capítulo 3, baseado em (HEFEZ; VILLELA, 2002) e em (MILIES, 2009), serão apresentados alguns conceitos da métrica de Hamming, lemas e propriedades que indicarão a quantidade de erros que podem ser detectados e corrigidos por um código; essa capacidade de correção está relacionada à quantidade de dígitos de redundância que serão acrescentados.

A classe dos códigos que é mais utilizada para detecção e correção de erros é a classe dos códigos lineares, que será apresentada no Capítulo 4, baseado em (HEFEZ; VILLELA, 2002), nele estão formas de codificar as palavras, verificar se a palavra recebida pertence ou não ao código, e um algoritmo para corrigir erros que estejam dentro da capacidade de correção do código.

Por fim, será apresentada uma proposta pedagógica, no Capítulo 5, com sugestão de

aplicação no Ensino Fundamental, utilizando recursos presentes no cotidiano dos alunos, tudo de forma simples e coerente com o nível fundamental de ensino.

CAPÍTULO

2

DÍGITOS VERIFICADORES PRESENTES NO COTIDIANO

O uso de códigos numéricos é cada vez mais comum no cotidiano, eles aparecem na identificação de produtos (códigos de barras), nos códigos postais, nos documentos pessoais, tais como Registro Geral (RG), Título de Eleitor, Cadastro de Pessoa Física (CPF), Carteira Nacional de Habilitação (CNH), entre outros.

O uso de códigos numéricos é vantajoso, pois pode ser entendido em todos os idiomas e possibilita registrar uma quantidade maior de informação do que se utilizássemos apenas nomes, por exemplo, se alguém se chama "José da Silva" e algum membro do governo precisa localizá-lo, provavelmente haverá vários outros homônimos, tornando difícil a diferenciação de cada um deles, entretanto o número do CPF é único e o "José da Silva" correto, será facilmente identificado. Por outro lado, é mais difícil perceber, visualmente, erros de digitação em códigos numéricos, por exemplo, se alguém digita o CPF "213.453.966-98", como podemos verificar se houve erro de digitação ou não? Com palavras é mais fácil de visualizar; se for digitada a palavra "sabpnete", por exemplo, é fácil perceber que há um erro de digitação e que a palavra correta era "sabonete".

Pensando nisso, foram criados dígitos verificadores de erros que permitem verificar, na maioria das vezes, se houve erro de digitação nos códigos numéricos, esses dígitos verificadores são resultados de operações matemáticas com os demais dígitos do código. Mostraremos nas próximas seções, como são calculados os dígitos verificadores presentes no CPF, nos códigos de barras e no sistema ISBN (International Standard Book Number), esses códigos estão associados ao sistema EAN (European Article Number), norma que garante o reconhecimento do código em todos os países, nesse sistema os códigos possuem de oito a treze algarismos.

2.1 Cadastro de Pessoa Física (CPF)

"CPF é um banco de dados gerenciado pela Secretaria da Receita Federal do Brasil - RFB que armazena informações cadastrais de contribuintes obrigados à inscrição no CPF, ou de cidadãos que se inscreveram voluntariamente." (RFB, 2015)

Figura 1 – CPF

Fonte: RFB (2015).

Seu código é da forma EAN-11, composto por onze algarismos, dos quais:

- Os oito primeiros algarismos representam o número-base.
- O nono algarismo indica a unidade da Federação em que o CPF foi cadastrado.
- Os dois últimos algarismos são dígitos verificadores.

A relação do nono algarismo com a unidade da Federação em que foi realizado o cadastro está representada na figura 2.

Os dígitos verificadores são obtidos através dos algarismos anteriores, se eles estiverem em desacordo com a lei de formação, significa que o número de CPF é inválido, ou seja, há erro no número apresentado. Por outro lado, se os dígitos verificadores estiverem de acordo com a lei de formação, então o número de CPF é válido, o que não significa, necessariamente, que haja uma pessoa cadastrada com ele no banco de dados da Receita Federal.

Seja $X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}$ um número de CPF em que X_i representa o algarismo de posição i, com i variando de 1 a 11, a escolha dos dígitos verificadores é dada pelas regras:

Cálculo do primeiro dígito verificador:

- Multiplica-se os nove primeiros algarismos (ordenados da esquerda para a direita) pelos números (1, 2, 3, 4, 5, 6, 7, 8, 9), ou seja, multiplica-se X_1 por 1, X_2 por 2, X_3 por 3, ..., X_9 por 9.
- Soma-se os resultados obtidos pelas multiplicações.

Figura 2 – Unidades da Federação

CÓDIGO EAN-11 PARA AS UNIDADES DA FEDERAÇÃO	
BRASIL	
0	Rio Grande do Sul
1	Distrito Federal, Goiás, Mato Grosso, Mato Grosso do Sul e Tocantins
2	Acre, Amapá, Amazonas, Pará, Rondônia e Roraima
3	Ceará, Maranhão e Piauí
4	Alagoas, Paraíba, Pernambuco e Rio Grande do Norte
5	Bahia e Sergipe
6	Minas Gerais
7	Espírito Santo e Rio de Janeiro
8	São Paulo
9	Paraná e Santa Catarina

Fonte: Fini (2009, p. 74).

- Divide-se a soma obtida por 11.
- O resto dessa divisão será o primeiro dígito verificador, ou seja, será o valor de X_{10}

Observação 1. Se o resto obtido for 10, o dígito verificador será 0.

Cálculo do segundo dígito verificador:

- Multiplica-se os dez primeiros algarismos (ordenados da esquerda para a direita), incluindo agora o primeiro dígito verificador obtido no passo anterior, pelos números (0, 1, 2, 3, 4, 5, 6, 7, 8, 9), ou seja, multiplica-se X_1 por 0, X_2 por 1, X_3 por 2, ..., X_{10} por 9.
- Soma-se os resultados obtidos pelas multiplicações.
- Divide-se a soma obtida por 11.
- O resto dessa divisão será o segundo dígito verificador, ou seja, será o valor de X_{11}

Observação 2. Se o resto obtido for 10, o dígito verificador será 0.

Exemplo 1. Vamos analisar o número "213.453.966-98", apresentado na introdução deste capítulo, e verificar se os dígitos verificadores estão em concordância com as regras apresentadas, caso não estejam, vamos encontrar os dígitos corretos. Como o nono dígito é igual a 6, podemos afirmar que esse cadastro foi realizado no estado de Minas Gerais.

Para o primeiro dígito verificador,temos:

$$(2\times 1)+(1\times 2)+(3\times 3)+(4\times 4)+(5\times 5)+(3\times 6)+(9\times 7)+(6\times 8)+(6\times 9)=$$

$$=2+2+9+16+25+18+63+48+54=237$$

$$237=(11\times 21)+6$$

Logo, o primeiro dígito verificador é igual a 6.

Para o segundo dígito verificador,temos:

$$(2\times 0)+(1\times 1)+(3\times 2)+(4\times 3)+(5\times 4)+(3\times 5)+(9\times 6)+(6\times 7)+(6\times 8)+(6\times 9)=$$

$$=0+1+6+12+20+15+54+42+48+54=252$$

$$252=(11\times 22)+10$$

Logo, o segundo dígito verificador é igual a 0.

Portanto, o número “213.453.966-98” não representa um CPF válido, o código correto deveria ser “213.453.966-60”.

2.2 Códigos de barras

O código de barras é da forma EAN-13, composto por 13 algarismos dos quais os três primeiros representam o país de origem, o último é um dígito verificador e os intermediários identificam o código da empresa fabricante e o código do produto.

Figura 3 – Código de Barras

Fonte: http://http://goo.gl/htsZ5m Acesso em 02 nov. 2015

Na figura 4 apresentamos uma tabela com os códigos identificadores de alguns países, o código referente ao Brasil é “789”.

O último algarismo, que corresponde ao dígito verificador, é gerado automaticamente por meio de operações matemáticas com os algarismos anteriores, mostraremos agora como é calculado o dígito verificador dos códigos de barras.

Figura 4 – Código EAN-13 de Alguns Países

CÓDIGO EAN-13 DE ALGUNS PAÍSES			
CÓDIGO	PAÍS	CÓDIGO	PAÍS
00 a 13	USA e Canadá	690 a 693	China
30 a 37	França	729	Israel
400 a 440	Alemanha	743	Nicarágua
45 a 49	Japão	744	Costa Rica
480	Filipinas	750	México
485	Armênia	770	Colômbia
528	Líbano	773	Uruguai
539	Irlanda	779	Argentina
560	Portugal	780	Chile
57	Dinamarca	789	**Brasil**
619	Tunísia	80 a 83	Itália
628	Arábia Saudita	84	Espanha
977	Periódicos (ISSN)	978 a 979	Livros (ISBN)

Fonte: Fini (2009, p. 73).

Seja $X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}X_{12}X_{13}$ um número de código de barras em que X_i representa o algarismo de posição i, com i variando de 1 a 13, a escolha do dígito verificador é dada pelas regras:

- Multiplica-se os doze primeiros algarismos (ordenados da esquerda para a direita) pelos números (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3), ou seja, multiplica-se X_1 por 1, X_2 por 3, X_3 por 1, ..., X_{12} por 3.
- Soma-se os resultados obtidos pelas multiplicações.
- Se a soma obtida for um número divisível por 10, o dígito verificador será “0”.
- Se a soma obtida não for um número divisível por 10, a diferença de 10 pelo resto da divisão efetuada será o dígito verificador.

Exemplo 2. Vamos analisar o número “7898357417892”, apresentado no início desta seção, e verificar se o dígito verificador está em concordância com as regras apresentadas.

Como o código formado pelos três primeiros dígitos é igual a 789, podemos afirmar que o produto foi fabricado no Brasil.

Para o dígito verificador,temos:

$$(7\times 1)+(8\times 3)+(9\times 1)+(8\times 3)+(3\times 1)+(5\times 3)+(7\times 1)+(4\times 3)+(1\times 1)+$$

$$+(7\times 3)+(8\times 1)+(9\times 3)=7+24+9+24+3+15+7+12+1+21+8+27=158$$

$$158=(10\times 15)+8$$

Como a soma não é um número divisível por 10, façamos a diferença de 10 pelo resto da divisão:

$$10-8=2$$

Logo, o dígito verificador do código de barras é "2" e está de acordo com a figura apresentada.

2.3 International Standard Book Number (ISBN)

> O ISBN - International Standard Book Number - é um sistema internacional padronizado que identifica numericamente os livros segundo o título, o autor, o país, a editora, individualizando-os inclusive por edição. Utilizado também para identificar software, seu sistema numérico é convertido em código de barras, o que elimina barreiras linguísticas e facilita a circulação e comercialização das obras. (ISBN, 2015)

Os códigos do sistema ISBN são da forma EAN-13, com treze algarismos que trazem informações sobre o título do livro, editora, prefixo EAN e dígito verificador, conforme figura abaixo:

Figura 5 – Código ISBN

Fonte: ISBN (2015).

O último algarismo, que corresponde ao dígito verificador, é gerado automaticamente, as regras para o cálculo são as mesmas aplicáveis aos códigos de barras, logo:

Seja $X_1X_2X_3X_4X_5X_6X_7X_8X_9X_{10}X_{11}X_{12}X_{13}$ um número do sistema ISBN em que X_i representa o algarismo de posição i, com i variando de 1 a 13, a escolha do dígito verificador é dada pelas regras:

- Multiplica-se os doze primeiros algarismos (ordenados da esquerda para a direita) pelos números (1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3), ou seja, multiplica-se X_1 por 1, X_2 por 3, X_3 por 1, ..., X_{12} por 3.

- Soma-se os resultados obtidos pelas multiplicações.
- Se a soma obtida for um número divisível por 10, o dígito verificador será “0”.
- Se a soma obtida não for um número divisível por 10, a diferença de 10 pelo resto da divisão efetuada será o dígito verificador.

Exemplo 3. Vamos verificar se o dígito verificador do código “978-85-333-0400-5” está correto.

Para o dígito verificador, temos:

$$(9\times 1)+(7\times 3)+(8\times 1)+(8\times 3)+(5\times 1)+(3\times 3)+(3\times 1)+(3\times 3)+(0\times 1)+$$
$$+(4\times 3)+(0\times 1)+(0\times 3)=9+21+8+24+5+9+3+9+0+12+0+0=100$$

$$100\div 10=10$$

Como a soma é um número divisível por 10, o dígito verificador correto é “0”, isso significa que o código ISBN apresentado não é válido, o código correto deveria ser “978-85-333-0400-0”.

CAPÍTULO

3

CÓDIGOS CORRETORES DE ERROS

3.1 Códigos

Vamos apresentar os elementos básicos de um código:

- Alfabeto, que representaremos por A é um conjunto finito cujos elementos são todos os símbolos que utilizaremos para formar as palavras, representaremos por "$q = |A|$" a quantidade de elementos de A, sendo chamado de código q-ário. Podemos citar como exemplo os códigos binários, $A = \{0,1\}$, e os códigos ternários, $A = \{0,1,2\}$.
- Palavras são sequências finitas de símbolos do alfabeto A.
- Comprimento é o número de letras, que representaremos por n, de uma palavra. Para facilitar a construção de um sistema de verificação e correção de erros, convencionaremos que todas as palavras tenham o mesmo comprimento, isso é sempre possível, basta adicionarmos símbolos neutros às palavras.
- Um código q-ário C, composto por palavras, de nossa escolha, com comprimento n, é um subconjunto de A^n, ou seja, dentre todas as palavras de comprimento n que podem ser formadas com o alfabeto A, escolhemos algumas que possuirão sentido.

O exemplo de código mais simples que podemos imaginar é a Língua Portuguesa, cujo alfabeto é composto por 26 letras mais o espaço, que consideraremos como uma letra, nossa maior palavra, "pneumoultramicroscopicossilicovulcanoconiótico", é composta por 46 letras, se acrescentássemos espaços à esquerda de todas as outras palavras da língua poderíamos deixar todas com comprimento 46.

Entretanto, nosso idioma não é bom para detectar e corrigir erros, por exemplo, se enviássemos a palavra "árvore" e o receptor recebesse a palavra "árvpre", seria fácil perceber

que foi cometido um erro e que a palavra mais próxima é “árvore”, já se a palavra enviada fosse “pato” e a recebida fosse “aato”, podemos detectar que há erro, mas como saber se a palavra correta é “rato”, “gato”, “jato”, “pato”, “mato”, ou tantas outras? Ou ainda, se a palavra recebida fosse “rato”, como seria possível identificar o erro? Uma solução para problemas como esse consiste em usar códigos numéricos que veremos mais adiante.

Sempre que enviamos uma mensagem estamos sujeitos a interferências eletromagnéticas, chamadas de ruídos, que podem causar erros, um mecanismo que possibilita detectar e corrigir esses erros consiste em codificar a mensagem acrescentando redundâncias, semelhante aos dígitos verificadores que foram estudados no capítulo anterior. Tal teoria será apresentada no Capítulo 4.

3.2 Métrica de Hamming

Definição 1. Dados dois elementos u e $v \in A^n = \underbrace{A \times A \times \cdots \times A}_{n \text{ vezes}}$, a distância de Hamming entre u e v é definida como

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

Exemplo 4. Em $\{0,1\}^4$, temos

$$d(0101,1001) = 2$$

$$d(1000,1111) = 3$$

$$d(1100,1101) = 1$$

$$d(0000,1111) = 4$$

$$d(1101,1101) = 0$$

Como veremos na proposição abaixo, a distância de Hamming satisfaz as três propriedades de métrica e chamaremos de métrica de Hamming.

Proposição 1. Dados u, v e $w \in A^n$, valem as seguintes propriedades:

1. Positividade: $d(u,v) \geq 0$, a igualdade só vale se, e somente se, $u = v$.
2. Simetria: $d(u,v) = d(v,u)$.
3. Desigualdade Triangular: $d(u,v) \leq d(u,w) + d(w,v)$.

Demonstração. (1) e (2) seguem imediatamente da definição.

(3) A contribuição das i-ésimas coordenadas de u e v para $d(u,v)$ é igual a zero se $u_i = v_i$, e igual a um se $u_i \neq v_i$.

No caso em que a contribuição é zero, certamente a contribuição das i-ésimas coordenadas a $d(u,v)$ é menor ou igual a das i-ésimas coordenadas a $d(u,w)+d(w,v)$, que será igual a 0, 1 ou 2.

No caso em que a contribuição é um, temos que $u_i \neq v_i$ e, portanto, não podemos ter $u_i = w_i$ e $w_i = v_i$. Por consequência, temos que a contribuição das i-ésimas coordenadas a $d(u,w)+d(w,v)$ é maior ou igual a 1 que é a contribuição das i-ésimas coordenadas a $d(u,v)$. □

Vamos definir agora o conceito de disco e esfera.

Definição 2. Sejam $a \in A^n$ e um número real $r > 0$, definimos o disco e a esfera de centro em a e raio r como sendo os respectivos conjuntos

$$D(a,r) = \{u \in A^n : d(u,a) \leq r\},$$

$$S(a,r) = \{u \in A^n : d(u,a) = r\}.$$

Disco e esfera são conjuntos finitos, veremos um lema que mostrará as suas cardinalidades que representaremos por $|D(a,r)|$ e $|S(a,r)|$, respectivamente.

Lema 1. Para todo $a \in A^n$ e todo número natural $r \geq 0$, temos que

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

em que n é o comprimento das palavras e q é a quantidades de letras presentes no alfabeto.

Demonstração. Primeiro é preciso notar que $S(a,i) \cap S(a,j) = \emptyset$ quando $i \neq j$, com isso fica fácil perceber que $D(a,r) = \bigcup_{i=0}^{r} S(a,i)$. Para finalizar, bastar provar que $|S(a,i)| = \binom{n}{i}(q-1)^i$ De fato, como temos q letras presentes no alfabeto, tomada qualquer letra de a, temos exatamente $(q-1)$ diferentes possibilidades de substituí-la para formar uma palavra diferente de a. Se quisermos que a distância da nova palavra a a seja i, teremos que fazer i alterações em a, ou seja, pelo princípio multiplicativo, temos $(q-1)^i$ possibilidades. Finalmente, para esgotar todas as possibilidades, levando em conta a posição de cada letra, é preciso multiplicar o resultado por $\binom{n}{i}$, logo

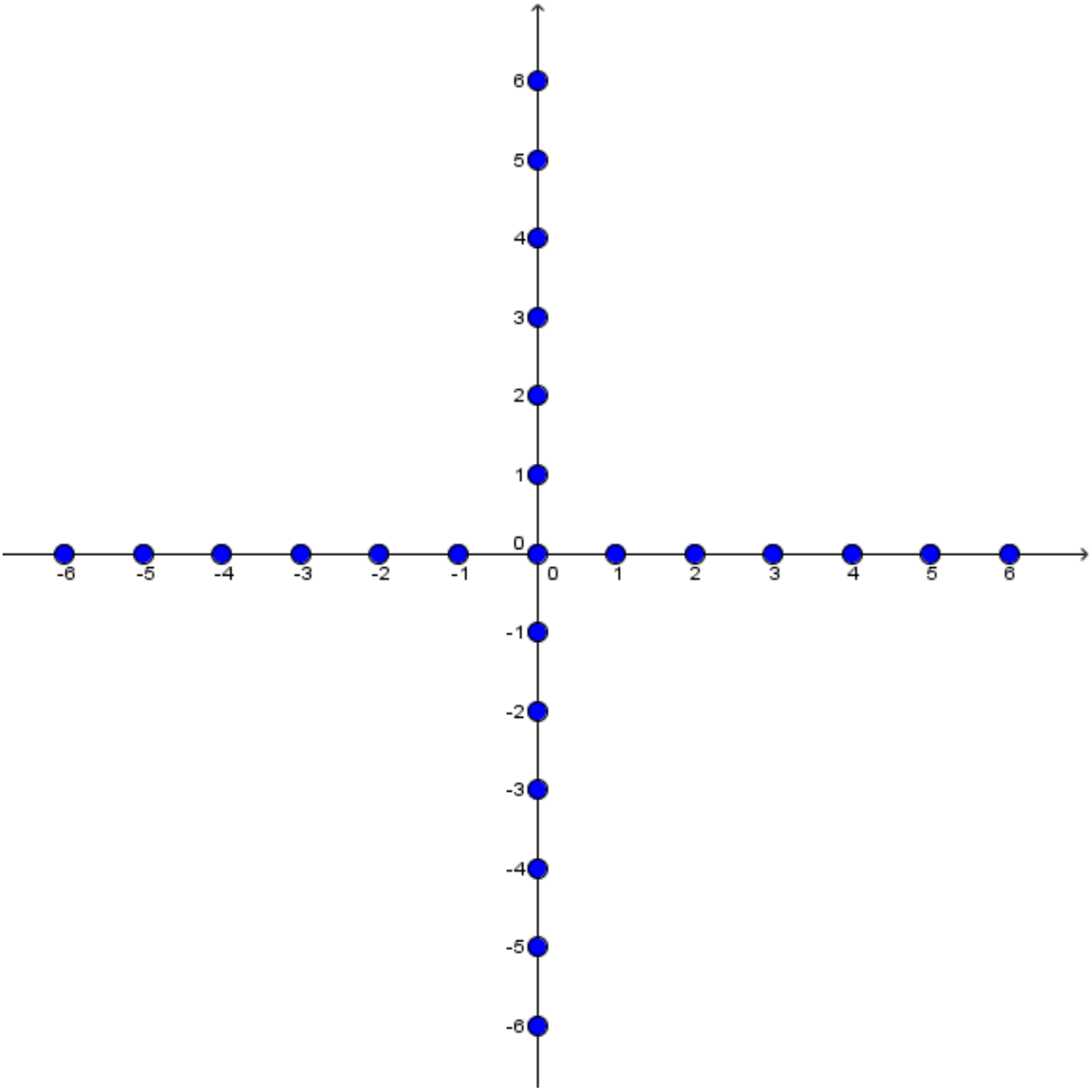
$$|S(a,i)| = \binom{n}{i} (q-1)^i.$$

□

É interessante notar que o disco e a esfera adotados na métrica de Hamming são diferentes de discos e esferas convencionais, vejamos alguns exemplos gráficos da métrica de Hamming aplicada no A^2 em que $A = \{-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6\}$.

Exemplo 5. Seja $v = (0,0)$. Todos os pares ordenados cuja distância de Hamming a v é menor ou igual a 1, devem ser da forma $(0,y)$ ou da forma $(x,0)$, conforme a figura 6.

Figura 6 – Disco de centro $v = (0,0)$ e raio $r = 1$



Fonte: Elaborada pelo autor.

Exemplo 6. Seja $v = (1,1)$ Todos os pares ordenados cuja distância de Hamming a v é menor ou igual a 1, devem ser da forma $(1,y)$ ou da forma $(x,1)$, conforme a figura 7.

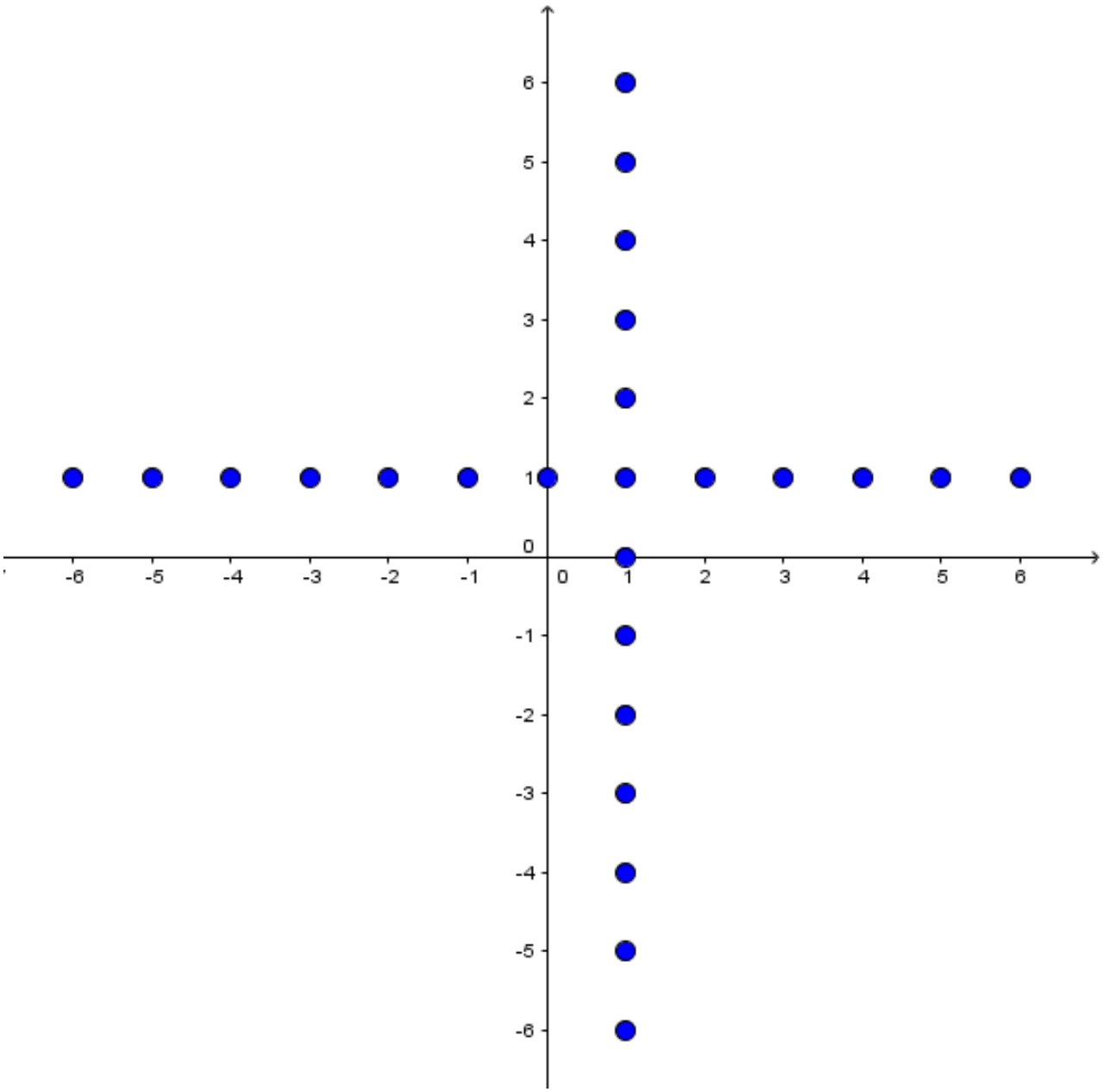
Definição 3. Seja C um código, a distância mínima de C é o número

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

Lema 2. Seja C um código com distância mínima d. Se c e c' são palavras distintas de C, então

$$D(c,\kappa) \cap D(c',\kappa) = \emptyset.$$

Figura 7 – Disco de centro $v = (1,1)$ e raio $r = 1$



Fonte: Elaborada pelo autor.

Aqui κ é definido por

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

onde $[x]$ denota a parte inteira de x.

Demonstração. Vamos supor, por absurdo, que existe t pertencente a $D(c,\kappa) \cap D(c',\kappa)$, então $d(t,c) \leq \kappa$ e $d(t,c') \leq \kappa$. Pelas propriedades de simetria e desigualdade triangular, temos

$$d(c,c') \leq d(c,t) + d(t,c') \leq 2\kappa \leq d-1$$

o que é um absurdo, pois $d(c,c') \geq d$. Logo $D(c,\kappa) \cap D(c',\kappa) = \emptyset$. □

A distância mínima d de um código é extremamente importante, pois a partir dela podemos definir a quantidade de erros que podem ser detectados e a quantidade de erros que podem ser corrigidos, conforme o teorema a seguir.

Teorema 1. Seja C um código com distância mínima d. Então C pode corrigir até $\kappa = \left[\frac{d-1}{2}\right]$ erros e detectar no máximo $d-1$ erros.

Demonstração. Suponha que ao transmitirmos uma palavra c do código cometemos t erros com $t \leq \kappa$, recebendo a palavra r, então $d(r,c) = t \leq \kappa$. Pelo Lema 2, $r \notin D(c', \kappa)$, para toda palavra $c' \neq c$. Isso determina c univocamente a partir de r.

Por outro lado, dada uma palavra do código, sempre que forem cometidos até $d-1$ erros, a palavra recebida não coincidirá com outra palavra do código, e assim, a detecção do erro será possível. □

Exemplo 7. Seja um código C, com distância mínima $d = 8$, então C detecta até $8-1 = 7$ erros e corrige no máximo $\kappa = \left[\frac{8-1}{2}\right] = 3$ erros.

Como a distância mínima está ligada à capacidade de detectar e corrigir erros, quanto maior for essa distância, maior será a quantidade de erros que poderão ser detectados e corrigidos, é fundamental poder calcular d ou pelo menos determinar uma cota superior.

CAPÍTULO

4

CÓDIGOS LINEARES

A classe dos códigos mais utilizadas é a classe dos códigos lineares, introduzida a seguir.

Definição 4. Seja K um corpo[1] finito com q elementos. Um código $C \subset K^n$ é dito linear se:

1. $(000\cdots0) \in C$;
2. dados $u_1, u_2 \in C$, $u_1 - u_2 \in C$;

Exemplo 8. Tomando o corpo $K = \mathbb{Z}_2 = \{[0],[1]\}$, tem-se que $\mathbb{Z}_2^5$ também é um corpo e que $C = \{(00000),(01011),(10110),(11101)\} \subset \mathbb{Z}_2^5$ é um código linear. [2]

Dada uma matriz $G = (I_k|A_{k\times(n-k)})_{k\times n}$ com entradas num corpo K, podemos considerar a aplicação dada por

$$T_G : K^k \to K^n$$
$$u = (x_1\cdots x_k) \mapsto T_G u = uG = (x_1\cdots x_k, uA).$$

Proposição 2. $C = ImT_G$ é um código linear.

Demonstração. É claro que:

1. $(000...0) \in ImT_G$.
2. Dados $v_1, v_2 \in ImT_G$, existem $u_1, u_2 \in K^n$ tais que $T_G u_1 = v_1$ e $T_G u_2 = v_2$. Assim, existe $u = u_1 - u_2 \in K^k$ tal que $T_G(u) = T_G(u_1 - u_2) = T_G u_1 - T_G u_2 = v_1 - v_2$.

□

[1] Ver Anexo A.
[2] Aqui e no que segue usaremos a notação $k = [k], k = 0, 1$.

Definição 5. Se C é um código linear, com $C = ImT_G$, a matriz $G = (I_k|A_{k\times(n-k)})_{k\times n}$ com entradas num corpo K, é dita matriz de codificação na forma padrão de C.

Nos próximos dois exemplos, vamos apresentar uma matriz geradora $G = (I_k|A_{k\times(n-k)})_{k\times n}$, na forma padrão e o código linear associado C.

Exemplo 9. Considere a matriz na forma padrão

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

e a aplicação dada por

$$T_G : K^2 \to K^5$$
$$u = (x_1\ x_2) \mapsto T_G u = uG = (x_1\ x_2\ x_1\ (x_1+x_2)\ (x_2)).$$

Tomando $K = \mathbb{Z}_2$, temos que $C = ImT_G = \{(00000),(01011),(10110),(11101)\} \subset \mathbb{Z}_2^5$ é o código linear associado.

Exemplo 10. Considere a matriz na forma padrão

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

e a aplicação dada por

$$T_G : K^3 \to K^6$$
$$u = (x_1\ x_2\ x_3) \mapsto T_G u = uG = (x_1\ x_2\ x_3\ (x_1+x_2)\ (x_1+x_3)\ (x_2+x_3)).$$

Tomando $K = \mathbb{Z}_2$, temos que

$$C = ImT_G = \{(000000),(001011),(010101),(100110),(011110),(101101),(110011),(111000)\} \subset \mathbb{Z}_2^6$$

é o código linear associado.

Definição 6. A transposta da matriz $A = [a_{i,j}]_{i,j=1}^{m,n}$ é a matriz $A^t = [a_{j,i}]_{j,i=1}^{n,m}$

A seguinte definição será utilizada para detectar se uma palavra pertence ou não ao código.

Definição 7. Seja $G = (I_k|A_{k\times(n-k)})_{k\times n}$ uma matriz de codificação na forma padrão, define-se a matriz teste de paridade por $H = (-A^t|I)_{(n-k)\times n}$.

Proposição 3. Sejam $G = (I_k|A_{k\times(n-k)})_{k\times n}$ uma matriz geradora do código $C = Im(T_G)$, escrita na forma padrão, e H a matriz teste de paridade do código C. Então $v = (v_1...v_n) \in C$ se, e somente se, $Hv^t = 0$.

Demonstração. ($\Rightarrow$) Suponha que $v \in Im(f_G)$. Logo existe $u \in \mathbb{R}^k$ tal que

$$v = uG = (u_1 ... u_k)(I_k|A) = (u, uA).$$

Como

$$v^t = \begin{pmatrix} u^t \\ \\ (uA)^t \end{pmatrix}_{n\times 1},$$

logo

$$Hv^t = (-A^t|I) \begin{pmatrix} u^t \\ \\ (uA)^t \end{pmatrix} = -A^t u^t + (uA)^t = -(uA)^t + (uA)^t = 0.$$

($\Leftarrow$) Suponha que $Hv^t = 0$ em que $v = (z, w)$, com $z \in \mathbb{R}^k$ e $w \in \mathbb{R}^{n-k}$. Assim

$$Hv^t = (-A^t|I) \begin{pmatrix} z^t \\ \\ w^t \end{pmatrix}_{n\times 1} = -A^t z^t + w^t = 0.$$

Logo $w^t = A^t z^t$, i.e., $w = zA$. Portanto, $v = (z, w) = (z, zA) = zG \in C$. □

Exemplo 11. Seja um código linear C gerado pela matriz

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

e com matriz teste de paridade

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

Sejam os vetores $v = (11101)$ e $v' = (10111)$, temos:

- $Hv^t = (000)^t$, logo $v \in C$.
- $Hv'^t = (001)^t$, logo $v' \notin C$.

Definição 8. Seja $x \in K^n$, o peso de x é o número inteiro

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

Em outras palavras, $\omega(x) = d(x, 0)$, em que d é a métrica de Hamming.

Definição 9. Define-se o peso de um código linear C como sendo o número inteiro

$$\omega(C) := \min\{\omega(x) : x \in C \setminus \{0\}\}.$$

Proposição 4. Seja $C \in K^n$ um código linear com distância mínima d. Temos que

1. Para todo x, y pertencente a K^n, $d(x,y) = \omega(x-y)$.
2. $d = \omega(C)$.

Demonstração. O item (1) segue diretamente das definições de métrica de Hamming e da de peso de um elemento. Para o item (2), temos que, como C é um código linear, para todo par de elementos x, y em C com $x \neq y$, temos que $z = x - y$ pertence a $C \setminus \{0\}$ e $d(x,y) = \omega(z)$. □

Exemplo 12. Tomando o código linear $C = \{(00000), (01011), (10110), (11101)\} \subset \mathbb{Z}_2^5$, temos

$$\omega(00000) = 0;$$

$$d(01011, 00000) = \omega(01011) = 3;$$

$$d(10110, 00000) = \omega(10110) = 3;$$

$$d(11101, 00000) = \omega(11101) = 4;$$

$$d(01011, 10110) = \omega(01011 - 10110) = 4;$$

$$d(01011, 11101) = \omega(01011 - 11101) = 3;$$

$$d(10110, 11101) = \omega(10110 - 11101) = 3;$$

$$\omega(C) = 3.$$

Logo, $d = 3 = \omega(C)$.

A proposição anterior é importante, pois fornece outra forma de calcular a distância mínima de um código, se inicialmente era preciso tomar duas a duas todas as M palavras do código e calcular a distância entre elas, efetuando assim $\binom{M}{2}$ cálculos, agora, com a equivalência de distância mínima e peso do código, basta calcular o peso de todas as palavras, exceto $(00...00)$, efetuando assim $M-1$ cálculos, o que gera um custo computacional bem menor que o inicial.

4.1 Decodificação

Definição 10. Dados um código C com matriz teste de paridade H e um vetor $v \in K^n$, define-se o vetor Hv^t como sendo a *síndrome* de v.

Definição 11. A diferença entre o vetor recebido r e o vetor transmitido c é chamada de vetor erro e, ou seja

$$e = r - c.$$

Exemplo 13. Se, em um determinado código $C \subset \mathbb{Z}_2^8$, tenha sido transmitida a palavra (10101010) e tenha sido recebida a palavra (00101011), temos

$$e = (10101010) - (00101011) = (10000001).$$

É importante notar que o número de erros presente na palavra recebida é igual ao peso do vetor erro.

Seja H a matriz teste de paridade do código, como $Hc^t = 0$, temos o seguinte resultado

$$He^t = H(r^t - c^t) = Hr^t - Hc^t = Hr^t - 0 = Hr^t.$$

Isso significa que o vetor erro tem a mesma síndrome da palavra recebida.

Chamemos de h^i a i-ésima coluna de H e $e = (\alpha_1 ... \alpha n)$ o vetor erro, então

$$\sum_{i=1}^{n} \alpha_i h^i = He^t = Hr^t.$$

Definição 12. Seja $v \in K^n$ e C um código linear, então definimos o conjunto

$$v + C = \{v + c : c \in C\}.$$

Lema 3. Os vetores u e v de K^n têm a mesma síndrome se, e somente se, $u \in v + C$.

Demonstração. $Hu^t = Hv^t \Leftrightarrow H(u-v)^t = 0 \Leftrightarrow u - v \in C \Leftrightarrow u \in v + C$. □

Na próxima proposição apresentaremos algumas propriedade que os conjuntos $v + C$ gozam.

Proposição 5. Seja C um (n,k)-código linear. Temos que

1. $v + C = v' + C \Leftrightarrow v - v' \in C$;
2. $(v + C) \cap (v' + C) \neq \emptyset \Leftrightarrow v + C = v' + C$;
3. $\bigcup_{v \in K^n} (v + C) = K^n$;

Demonstração. Vamos demonstrar somente o item 1, sendo que as demonstrações dos itens 2 e 3 são análogas.

Suponha que $v + C = v' + C$. Dado $x \in v + C = v' + C$, existem $c_1, c_2 \in C$ tais que $x = v + c_1 = v' + c_2$. Logo $v - v' = c_2 - c_1 \in C$.

Reciprocamente, suponha $v - v' \in C$. Se $x \in v + C$, então existe $c_1 \in C$ tal que $x = v + c_1$. Ainda como $x = v - v' + v' + c_1$ e $v - v' \in C$, segue que $x \in v' + C$. Portanto, temos que $v + C \subset v' + C$.

Por outro lado, Se $x \in v' + C$, então existe $c_2 \in C$ tal que $x = v' + c_2$. Ainda, como $x = v' - v + v + c_2$ e $v' - v \in C$, segue que $x \in v + C$. Portanto, temos que $v' + C \subset v + C$.

□

Definição 13. Define-se classe lateral de v segundo C como $v + C$.

Exemplo 14. Tomando o código linear $C = \{(00000), (01011), (10110), (11101)\} \subset \mathbb{Z}_2^5$,temos as classes laterais

$$(00000) + C = \{(00000), (01011), (10110), (11101)\};$$

$$(00001) + C = \{(00001), (01010), (10111), (11100)\};$$

$$(00010) + C = \{(00010), (01001), (10100), (11111)\};$$

$$(00100) + C = \{(00100), (01111), (10010), (11001)\};$$

$$(01000) + C = \{(01000), (00011), (11110), (10101)\};$$

$$(10000) + C = \{(10000), (11011), (00110), (01101)\};$$

$$(10001) + C = \{(10001), (11010), (00111), (01100)\};$$

$$\vdots$$

$$(11111) + C = \{(11111), (10100), (01001), (00010)\};$$

Com base no item (1), podemos afirmar

$$v + C = C \Leftrightarrow v \in C.$$

Definição 14. Um vetor peso mínimo numa classe lateral é chamado de elemento *líder* dessa classe.

Proposição 6. Seja C um código linear em K^n com distância mínima d. Se $u \in K^n$ é tal que

$$\omega(u) \leq \kappa = \left[\frac{d-1}{2}\right],$$

então u é o único elemento líder de sua classe.

Demonstração. Vamos supor que existam u e u', com $u \neq u'$, com $\omega(u) \leq \left[\frac{d-1}{2}\right]$ e $\omega(u') \leq \left[\frac{d-1}{2}\right]$ tais que u e u' pertençam à mesma classe de C. Logo $u - u' \in C$ e

$$\omega(u - u') \leq \omega(u) + \omega(u') \leq \left[\frac{d-1}{2}\right] + \left[\frac{d-1}{2}\right] \leq d - 1.$$

Pela Proposição 4 (1) $d(u, u') = \omega(u - u')$ e, por hipótese, $u - u' \in C$, temos que $d(u, u') \leq d - 1 < d$ o que é um absurdo, logo $u = u'$. □

Observação 3. Para encontrar os líderes de classe é preciso tomar todos os elementos u que satisfaçam $\omega(u) \leq \left[\frac{d-1}{2}\right] = \kappa$. A Proposição 6 nos garante que são líderes de uma e somente uma classe.

Para executar o algoritmo de decodificação que apresentaremos a seguir, primeiro é necessário montar uma tabela em que colocaremos todos os elementos $u \in K^n$ tais que $\omega(u) \leq \kappa$ acompanhados de suas respectivas síndromes, Hu^t, feito isso, podemos seguir os passos.

1. Calcular a síndrome s da palavra recebida r, ou seja $s^t = Hr^t$.
2. Comparar s com as síndromes da tabela.
3. Se s estiver na tabela, tomar ℓ, o elemento líder da classe determinada por s, e substituir r por $r - \ell$.
4. Se s não estiver na tabela, então na palavra recebida estão presentes mais do que κ erros.

Sejam r, c e e, respectivamente, a palavra recebida, a palavra transmitida e o vetor erro, como $Hr^t = He^t$, o algoritmo acima é válido, pois a síndrome de r determina a classe lateral em que e se encontra e, se $\omega(e) \leq \kappa$, a Proposição 6 nos garante que e é o único elemento ℓ líder de sua classe e está presente na tabela, bastando calcular $c = r - e = r - \ell$.

Exemplo 15. Seja um código linear $C \subset \mathbb{Z}_2^6$, com distância mínima $d = 3$, matriz teste de paridade H dada por

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

e capacidade de correção $\kappa = \left[\frac{3-1}{2}\right] = 1$. Vamos tomar todos os possíveis vetores cujo peso é menor ou igual a 1 e calcular suas respectivas síndromes.

Líder	Síndrome
000000	000
100000	110
010000	101
001000	011
000100	100
000010	010
000001	001

Suponha que tenham sido recebidas as palavras $r = (001011)$, $r' = (111101)$ e $r'' = (011001)$, vamos analisar o que acontece em cada caso.

- Para a palavra recebida $r = (001011)$, calculamos sua síndrome $Hr^t = (000)^t$, logo r pertence ao código e aceitamos r como sendo a palavra transmitida.

- Para a palavra recebida $r' = (111101)$, calculamos sua síndrome $Hr'^t = (101)^t$, logo $e = (010000)$ e a palavra transmitida é $r' - e = (111101) - (010000) = (101101)$.

- Para a palavra recebida $r'' = (011001)$, calculamos sua síndrome $Hr''^t = (111)^t$ que não aparece na tabela, logo, em r, foram cometidos mais do que κ erros.

Exemplo 16. Seja um código linear $C \subset \mathbb{Z}_2^9$, com distância mínima $d = 6$, matriz teste de paridade H dada por

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

e capacidade de correção $\kappa = \left[\frac{6-1}{2}\right] = 2$. Vamos tomar todos os possíveis vetores cujo peso é menor ou igual a 2 e calcular suas respectivas síndromes.

Líder	Síndrome	Líder	Síndrome
000000000	0000000	010100000	0010111
000000001	0000001	011000000	1110111
000000010	0000010	001000001	1000001
000000100	0000100	001000010	1000010
000001000	0001000	001000100	1000100
000010000	0010000	001001000	1001000
000100000	0100000	001010000	1010000
001000000	1000000	001100000	1100000
010000000	0110111	000100001	0100001
100000000	1101101	000100010	0100010
100000001	1101100	000100100	0100100
100000010	1101111	000101000	0101000
100000100	1101001	000110000	0110000
100001000	1100101	000010001	0010001
100010000	1111101	000010010	0010010
100100000	1001101	000010100	0010100
101000000	0101101	000011000	0011000
110000000	1011010	000001001	0001001
010000001	0110110	000001010	0001010
010000010	0110101	000001100	0001100
010000100	0110011	000000101	0000101
010001000	0111111	000000110	0000110
010010000	0100111	000000011	0000011

Suponha que tenham sido recebidas as palavras $r = (111011010)$, $r' = (011110101)$ e $r'' = (110000111)$, vamos analisar o que acontece em cada caso.

- Para a palavra recebida $r = (111011010)$, calculamos sua síndrome $Hr^t = (0000000)^t$, logo r pertence ao código e aceitamos r como sendo a palavra transmitida.
- Para a palavra recebida $r' = (011110101)$, calculamos sua síndrome $Hr'^t = (1000010)^t$, logo $e = (001000010)$ e a palavra transmitida é $r' - e = (011110101) - (001000010) = (010110111)$.
- Para a palavra recebida $r'' = (110000111)$, calculamos sua síndrome $Hr''^t = (1011101)^t$ que não aparece na tabela, logo, em r, foram cometidos mais do que κ erros.

CAPÍTULO

5

PROPOSTA DO TEMA

O público-alvo desta proposta são alunos da Educação Básica, mais especificamente, alunos do oitavo ano do Ensino Fundamental[1]. O tempo de aplicação é de quatro a cinco aulas com duração de cinquenta minutos cada.

Os pré-requisitos para os estudantes são conhecimentos do sistema decimal posicional e domínio das quatro operações.

A proposta se divide em cinco etapas e utilizará recursos presentes no cotidiano dos alunos, tais como CPF e código de barras.

5.1 Primeira etapa: Dígitos verificadores de erros no CPF

O professor deve solicitar aos alunos que levem anotado o número do CPF de duas pessoas diferentes.

Em um primeiro momento, o professor explicará aos alunos como é formado o número do CPF, especialmente o dígito que representa a unidade da Federação em que o CPF foi cadastrado e o processo de cálculo dos dígitos verificadores.

Cada aluno deverá aplicar o algoritmo do cálculo dos dígitos verificadores para os dois números de CPF que anotou, e verificar se eles estão corretos.

Feita a primeira etapa, cada aluno escreverá em um papel os números de CPF que levou, mas, omitirá os dígitos verificadores.

Os números de CPF serão trocados entre os alunos, o objetivo agora é que cada um calcule os dígitos verificadores dos CPFs que receberam para, em seguida, conferir com os números originais, se aplicaram o algoritmo corretamente, caso haja erros, o professor auxiliará cada aluno a identificar o erro cometido.

[1] A proposta também pode ser adaptada para aplicação no Ensino Médio.

5.2 Segunda etapa: Dígito verificador de erro no código de barras

O professor deve solicitar que os alunos levem três embalagens diferentes que possuam código de barras, se for possível, dentre essas três embalagens, duas devem ser da mesma marca, mas de produtos diferentes.

A atividade se inicia com o professor explicando aos alunos as informações contidas nos códigos de barras, indicando os algarismos referentes ao país de origem, a empresa que produziu o produto, o código do produto e o dígito verificador, para esse último, o professor mostrará qual o processo que deve ser aplicado para calcular o dígito verificador.

Cada aluno deverá observar o país de origem de cada produto, bem como reparar que produtos de mesma marca possuem parte dos códigos iguais.

Os alunos deverão escolher duas embalagens e aplicar o algoritmo para confirmar se o dígito verificador está correto.

Para finalizar essa etapa, o professor passará na lousa um número de código de barras com o dígito verificador omitido. Os alunos deverão identificar o país de origem do produto e calcular qual o dígito verificador que corresponde ao código.

5.3 Terceira etapa: Linguagem de computadores

O professor explicará aos alunos o conceito de números binários, a quantidade de algarismos utilizados, a relação entre a posição do algarismo e a potência de base 2 a que o algarismo deve ser multiplicado.

O próximo passo será construir com os alunos os quinze primeiros números do sistema binário, fazendo comparativos com o sistema decimal.

Na sequência, os alunos aprenderão sobre a operação de adição no sistema binário. Para facilitar esse aprendizado, o professor fará um comparativo com a adição no sistema decimal e usará os quinze primeiros números que foram construídos no passo anterior.

Para finalizar essa etapa, será apresentado aos alunos os códigos binários[2] correspondentes às letras do alfabeto e, para que possam visualizar como são as informações que o computador trabalha, cada aluno deverá escrever seu nome em linguagem computacional.

5.4 Quarta etapa: Noção de códigos corretores

Essa atividade será realizada com o auxílio do editor de textos *Word*.

[2] Esta tabela está presente no Anexo B

O professor explicará o conceito de distância de Hamming entre palavras e mostrará alguns exemplos simples.

O próximo passo será digitar no editor de textos a palavra “paralelepíprdo”, o editor acusará erro e mostrará como sugestão a palavra “paralelepípedo”.

Neste momento o professor questionará os alunos por que o editor não apresentou outra palavra como sugestão e fará a relação com a distância entre as palavras.

A próxima palavra a ser digitada será “ventilsdur” e o editor mostrará como sugestão a palavra “ventilador”, os alunos deverão analisar a distância entre as duas palavras e refletir por que a sugestão de correção foi única.

A última palavra a ser digitada será “hato”, o editor acusará erro, porém haverá mais de uma sugestão de correção, os alunos deverão refletir por que isso ocorre e relacionar com a distância entre as palavras.

Para finalizar, os alunos deverão comparar as palavras digitadas e refletir, com a mediação do professor, por que as duas primeiras palavras possuíam apenas uma sugestão, enquanto a última possuía várias? O computador poderia corrigir automaticamente todas as palavras? Se alguém enviasse a palavra “rato” e o destinatário recebesse a palavra “gato”, como identificaria o erro?

5.5 Quinta etapa: Códigos corretores

A teoria de Códigos Corretores de erro é uma ferramenta matemática desenvolvida para detectar e corrigir erros.

Utilizando o que foi apresentado nas etapas anteriores, o professor construirá com os alunos uma palavra em $\mathbb{Z}_2$ com três dígitos de informação e mais três dígitos de checagem, esta palavra será da forma: $(x_1;x_2;x_3;x_1+x_2;x_1+x_3;x_2+x_3)$.

Com uma breve explicação do professor sobre as somas utilizadas na construção dos dígitos de checagem ($x_n+x_k=0$ se a soma for par e $x_n+x_k=1$ se a soma for ímpar, com $1\leq n\leq 3$, $1\leq k\leq 3$ e $n\neq k$), os alunos deverão codificar todas as oito palavras possíveis do código.

Com as palavras em mãos, o professor questionará: “se fosse recebida a palavra ‘101101’, ela pertence ao código?” os alunos deverão responder que sim e concluir que a palavra recebida não contém erro.

Continuando com as verificações, o professor apresentará a palavra “011100 e os alunos deverão verificar que ela não pertence ao código, logo há erro na palavra transmitida, para corrigi-la, eles precisarão encontrar qual a palavra do código que possui a menor distância com a palavra recebida (desde que essa distância seja única), após os cálculos das distâncias, será

concluído que a palavra pode ser corrigida por “011110”.

A próxima palavra a ser apresentada será “111111”, que os alunos facilmente identificarão que a palavra contém erro, pois não pertence ao código, porém, nesse caso, ao calcularem as distâncias com as palavras do código, eles verão que não é possível corrigi-la, pois a menor distância com as palavras do código não é única.

Para finalizar a atividade, o professor explicará que o conteúdo trabalhado nessas atividades é o que acontece, claro que de forma mais avançada, quando enviamos um e-mail, ligamos para alguém que está em outro país, recebemos uma imagem pela televisão, entre outros.

5.6 Percepções pessoais

A proposta foi aplicada no 8ºA da Escola Municipal de Ensino Fundamental Professora Eponina de Britto Rossetto e as percepções do professor que aplicou são apresentadas a seguir.

Na primeira etapa, os alunos se mostraram interessados ao descobrir que o nono dígito de CPF representa a Unidade da Federação em que foi realizado o cadastro, eles fizeram, inclusive, comparações com os números levados e o estado de origem dos respectivos donos.

Outro fato que chamou a atenção dos estudantes foi descobrir que os dois últimos dígitos são obtidos por meio de operações matemáticas básicas envolvendo os algarismos anteriores, operações estas que, apesar de elementares, possuem uma grande utilidade no cotidiano que a grande maioria não sabia que existia.

Um fator fundamental para o bom desenvolvimento dessa etapa foi que os alunos puderam comprovar, com número de CPF próprio ou de parentes, que os dígitos verificadores realmente apresentam a relaçao ensinada.

Na segunda etapa, semelhante à primeira, os alunos se mostraram curiosos com o significado de cada grupo de dígitos do código de barras, fizeram comparações de códigos de produtos de mesma marca e produtos de marcas diferentes, perceberam que todos os códigos por eles levados se iniciam com o número “789”, indicando que o produto é fabricado no Brasil.

Tanto na primeira etapa, quanto na segunda, o professor deu dicas e estimulou que os alunos desenvolvessem a soma utilizando cálculo mental, em primeiro momento, eles apresentaram dificuldade nesse desenvolvimento, entretanto, com o passar do tempo, os estudantes passaram a dominar melhor o cálculo mental e a utilizá-lo com maior propriedade.

A terceira etapa se mostrou a mais trabalhosa de todo processo, alguns alunos entenderam muito rápido e conseguiram escrever mais do que os números solicitados, entretanto, outros apresentaram maior dificuldade em entender a correspondência existente entre números binários e decimais, necessitando de uma intervenção maior do professor.

Após escreverem os 15 primeiros números em base binária, o professor forneceu uma

folha que apresenta os códigos binários de todas as letras do alfabeto português, diferenciando maiúsculas e minúsculas, e solicitou que eles escrevessem o primeiro nome utilizando os códigos binários, essa parte foi realizada de forma autônoma e bem ágil.

Na quarta etapa os alunos conseguiram entender rapidamente o conceito de distância entre palavras e apresentaram conclusões sobre as palavras sugeridas pelo corretor ortográfico, associando-as com as distâncias entre elas.

Na quinta etapa os alunos apresentaram um pouco de dificuldade para entender o significado dos dígitos de redundância, mas não apresentaram dificuldade para construir o código proposto pelo professor e para calcular as distâncias das palavras apresentadas.

Com a mediação do professor, os estudantes puderam concluir que quando uma palavra é a única do código que possui a menor distância com a palavra recebida, então é possível corrigir o erro, mas quando é mais de uma palavra que possui a menor distância com a palavra apresentada, então o erro não pode ser corrigido.

REFERÊNCIAS

FINI, M. I. Controle dos códigos de identificação. **Revista do Professor – Atualidades**, p. 70–75, 2009. Disponível em: <http://docplayer.com.br/5199819-Controle-dos-codigos-de-identificacao.html>. Acesso em: 29 out. 2015. Citado 3 vezes nas páginas 19, 23 e 25.

HEFEZ, A.; VILLELA, M. L. T. **Códigos corretores de erros**. Rio de Janeiro: Instituto de Matematica Pura e Aplicada, 2002. Citado 3 vezes nas páginas 19, 59 e 61.

ISBN, A. B. do. **International Standard Book Number**. [S.l.], 2015. Disponível em: <http://www.isbn.bn.br/website/o-que-e-isbn>. Acesso em: 29 out. 2015. Citado na página 26.

MILIES, C. P. Breve introdução à teoria dos códigos corretores de erros. **Colóquio de Matemática da Região Centro-Oeste**, 2009. Disponível em: <http://http://www.sbm.org.br/docs/coloquios/CO-1-09.pdf>. Acesso em: 29 out. 2015. Citado na página 19.

RFB, R. F. do B. **Cadastro de Pessoa Física**. [S.l.], 2015. Disponível em: <http://www.receita.fazenda.gov.br/PessoaFisica/cpf/PerguntasRespostas/PerguntasRespostas.htm#1>. Acesso em: 1 nov. 2015. Citado na página 22.

APÊNDICE

A

EXERCÍCIOS SUGERIDOS

A seguir, apresentamos alguns exercícios que podem ser utilizados pelo professor em sala de aula

A.1 Capítulo 2

Exercício 1. Em cada item abaixo, descubra qual a Unidade da Federação foi realizado o cadastro do CPF, bem como os dois dígitos verificadores correspondentes:

a) 785.963.196-XY
b) 282.934.448-XY
c) 179.236.740-XY
d) 876.977.448-XY

Exercício 2. Com relação aos códigos de barras abaixo, descubra o país de origem do produto, bem como o dígito verificador correspondente:

a) 528734232957X
b) 619445623112X
c) 750776823109X
d) 789654819895X

Exercício 3. Em cada código ISBN abaixo, descubra se o dígito verificador está correto, caso não esteja, escreva qual deveria ser de modo a tornar o código válido:

a) 978-85-244-0312-5
b) 978-85-85818-08-7
c) 978-85-7542-643-8
d) 978-85-7600-352-6

A.2 Capítulo 3

Exercício 4. Dado o alfabeto $A = \{0,1\}$, escreva todas as palavras de comprimento $n = 4$.

Exercício 5. Calcule a distância de Hamming, nos inteiros, das palavras:

a) (110011) e (011001).

b) (2352) e (2253).

c) (22110) e (12000).

d) (11101) e (10101).

Exercício 6. Dado o alfabeto $A = \{0,1,2,3,4,5\}$, calcule a quantidade de elementos que pertencem:

a) Ao disco de centro (21113) e raio $r = 4$.

b) Ao disco de centro (0000) e raio $r = 3$.

c) Á esfera de raio (1225) e raio $r = 2$.

d) Á esfera de raio (443522) e raio $r = 4$.

Exercício 7. Seja C um código com distância mínima $d = 13$, qual a quantidade máxima de erros que podem ser detectados? Qual a quantidade máxima que pode ser corrigida?

Exercício 8. Calcule κ para códigos com distância mínima apresentada abaixo

a) $d = 4$.

b) $d = 1$.

c) $d = 2$.

d) $d = 6$.

A.3 Capítulo 4

Exercício 9. Calcule o peso das seguintes palavras:

a) (01111001).
b) (11001100011).
c) (11111111).
d) (0110000101).
e) (0000000).
f) (11010100011).
g) (100010000).
h) (110011001111).

Exercício 10. Seja um código linear $C \subset \mathbb{Z}_2^6$, com matriz geradora

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

Codifique todas as palavras de comprimento $n = 3$.

Exercício 11. Seja um código linear $C \subset \mathbb{Z}_2^6$, com matriz teste de paridade

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

Calcule a síndrome das palavras abaixo e diga se elas pertencem ou não ao código C.

a) (111111). e) (110010).
b) (101101). f) (111000).
c) (001101). g) (110110).
d) (101011). h) (010110).

Exercício 12. Seja um código linear $C \subset \mathbb{Z}_2^5$, com matriz teste de paridade

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

e distância mínima $d = 3$. Corrija, se possível, as seguintes palavras recebidas:

a) 11111. f) 11011.
b) 01000. g) 01111.
c) 10101. h) 10001.
d) 11010. i) 10000.
e) 11101. j) 00001.

Exercício 13. Seja um código linear $C \subset \mathbb{Z}_2^8$, com matriz teste de paridade

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

e distância mínima $d = 5$. Corrija, se possível, as seguintes palavras recebidas:

a) 10011110. f) 10101110.
b) 11110010. g) 10011111.
c) 01100001. h) 11111100.
d) 11111111. i) 11110011.
e) 11101101. j) 11111110.

A.4 Soluções

Exercício 1.
a) Minas Gerais, $X = 2,\ Y = 0$
b) São Paulo, $X = 0,\ Y = 0$
c) Rio Grande do Sul, $X = 6,\ Y = 0$
d) São Paulo, $X = 0,\ Y = 5$

Exercício 2.
a) Líbano, $X = 9$
b) Tunísia, $X = 6$
c) México, $X = 5$
d) Brasil, $X = 7$

Exercício 3.
a) Correto
b) 978-85-85818-08-1
c) 978-85-7542-643-2
d) Correto

Exercício 4. (0000), (0001), (0010), (0100), (1000), (0011), (0101), (0110), (1001), (1010), (1100), (0111), (1101), (1011), (1110), (1111)

Exercício 5. a) $d((110011),(011001)) = 3$.

b) $d((2352),(2253)) = 2$.

c) $d((22110),(12000)) = 3$.

d) $d((11101),(10101)) = 1$.

Exercício 6. a) 4651.

b) 671.

c) 150.

d) 9375.

Exercício 7. Podem ser detectados no máximo 12 erros e corrigidos no máximo 6 erros.

Exercício 8. a) $\kappa = 1$.

b) $\kappa = 0$.

c) $\kappa = 0$.

d) $\kappa = 2$.

Exercício 9.
a) $\omega(01111001) = 5$.
b) $\omega(11001100011) = 6$.
c) $\omega(11111111) = 8$.
d) $\omega(0110000101) = 4$.
e) $\omega(0000000) = 0$.
f) $\omega(11010100011) = 6$.
g) $\omega(100010000) = 2$.
h) $\omega(110011001111) = 8$.

Exercício 10. (000000), (001110), (010101), (100011), (011011), (101101), (110110), (111000)

Exercício 11.
a) (111). Não pertence.
b) (000). Pertence.
c) (011). Não pertence.
d) (110). Não pertence.
e) (101). Não pertence.
f) (000). Pertence.
g) (000). Pertence.
h) (011). Não pertence.

Exercício 12.
a) 11110.
b) 00000.
c) 10101.
d) 11110.
e) 10101.
f) 01011.
g) 01011.
h) 10101.
i) 00000.
j) 00000.

Exercício 13.

a) 10011110.
b) 11110011.
c) 01101101.
d) 11110011.
e) 01101101.
f) 10011110.
g) 10011110.
h) Não pode ser corrigido.
i) 11110011.
j) 10011110.

ANEXO

A

ANÉIS E CORPOS

Este anexo é baseado em (HEFEZ; VILLELA, 2002)

Definição 15. Um conjunto A é chamado de anel se ele é munido de duas operações,

$$+ : A \times A \to A$$
$$(a,b) \mapsto a+b$$

e

$$\cdot : A \times A \to A$$
$$(a,b) \mapsto a \cdot b$$

chamadas de adição e multiplicação, respectivamente, e além disso, devem satisfazer as seguintes propriedades:

1. Associatividade da adição:

$$\forall a,b,c \in A,\ (a+b)+c = a+(b+c).$$

2. Elemento neutro para a adição:

   Existe um elemento chamado zero e denotado por 0, tal que

$$\forall a \in A,\ a+0 = 0+a = a.$$

3. Elemento inverso para a adição:

   Dado $a \in A$, existe um elemento chamado simétrico de a e denotado por $-a$, tal que

$$a+(-a) = (-a)+a = 0.$$

4. Comutatividade da adição:

$$\forall a,b \in A,\ a+b=b+a.$$

5. Associatividade da multiplicação:

$$\forall a,b,c \in A,\ (a\cdot b)\cdot c = a\cdot(b\cdot c).$$

6. Elemento neutro para a multiplicação:

   Existe um elemento chamado unidade e denotador por 1, tal que

$$\forall a \in A,\ a\cdot 1 = 1\cdot a = a.$$

7. Distributividade da multiplicação em relação à adição:

$$\forall a,b,c \in A,\ a\cdot(b+c) = a\cdot b + a\cdot c.$$

   Se vale a comutatividade da multiplicação, isto é,

$$\forall a,b \in A,\ a\cdot b = b\cdot a,$$

o anel é dito comutativo.

Definição 16. Corpo é um anel em que todo elemento não nulo é invertível.

A.1 Classes residuais de inteiros

Chama-se classe residual de $\mathbb{Z}$ módulo m, a classe formada pelos restos das divisões dos inteiros por m, ou seja

$$\mathbb{Z}_m = \{[0],[1],...,[m-1]\},$$

se $i,j = 0,1,...,m-1$ e, se $i \neq j$, então $[i] \neq [j]$.

Dado $a \in \mathbb{Z}$, pelo algoritmo da divisão euclidiana, existem inteiros q e r univocamente determinados pelas condições $a = mq + r$ com $0 \leq r \leq m-1$, o que implica haver um único inteiro r com $0 \leq r \leq m-1$, tal que $[a] = [r]$.

Logo $\mathbb{Z}_m$ é um anel finito com exatamente m elementos.

Exemplo 17. Para $m = 2$, vamos montar as tabelas de adição e de multiplicação do anel $\mathbb{Z}_2 = \{[0],[1]\}$:

+	[0]	[1]
[0]	[0]	[1]
[1]	[1]	[0]

·	[0]	[1]
[0]	[0]	[0]
[1]	[0]	[1]

Como $[1]$ é o único elemento não nulo de $\mathbb{Z}_2$ e é invertível em relação à multiplicação, então $\mathbb{Z}_2$ é um corpo.

Exemplo 18. Para $m = 5$, vamos montar as tabelas de adição e de multiplicação do anel $\mathbb{Z}_5 = \{[0],[1],[2],[3],[4]\}$:

+	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[1]	[2]	[3]	[4]
[1]	[1]	[2]	[3]	[4]	[0]
[2]	[2]	[3]	[4]	[0]	[1]
[3]	[3]	[4]	[0]	[1]	[2]
[4]	[4]	[0]	[1]	[2]	[3]

·	[0]	[1]	[2]	[3]	[4]
[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]
[2]	[0]	[2]	[4]	[1]	[3]
[3]	[0]	[3]	[1]	[4]	[2]
[4]	[0]	[4]	[3]	[2]	[1]

Como $[1]$, $[2]$, $[3]$ e $[4]$ são invertíveis em relação à multiplicação, com inversos $[1]$, $[3]$, $[2]$ e $[4]$, respectivamente, então $\mathbb{Z}_5$ é um corpo.

Proposição 7. $[a] \in \mathbb{Z}_m$ é invertível se, e somente se, $\text{MDC}(a,m) = 1$.

Demonstração. Vamos supor que $[a]$ seja invertível. Logo, existe $[b] \in \mathbb{Z}$ tal que $[a] \cdot [b] = 1$. O que implica $[a \cdot b] = [1]$ e, consequentemente, $a \cdot b \equiv 1 \bmod m$, logo $m | a \cdot b - 1$ que é o mesmo que dizer que existe um inteiro s tal que

$$s \cdot m + a \cdot b = 1.$$

Por outro lado, $\text{MDC}(a,m)|a$ e $\text{MDC}(a,m)|m$ isso implica, pela equação acima, que $\text{MDC}(a,m)|1$. Logo $\text{MDC}(a,m) = 1$.

Reciprocamente, se o $\text{MDC}(a,m) = 1$, então existem inteiros b e c tais que $b \cdot a + c \cdot m = 1$ (para mais detalhes, veja (HEFEZ; VILLELA, 2002)). Logo, $b \cdot a \equiv 1 \bmod m$, o que implica $[a] \cdot [b] = [a \cdot b] = [1]$ e, por consequência, $[a]$ é invertível. □

Proposição 8. O anel $\mathbb{Z}_m$ é um corpo se, e somente se, m é um número primo.

Demonstração. $\mathbb{Z}_m$ será um corpo se, e somente se, todos os elementos $[1],[2],...,[m-1]$ forem invertíveis, entretanto, pela Proposição 7, isso só ocorrerá se $\text{MDC}(1,m) = \text{MDC}(2,m) = ... = \text{MDC}(m-1,m) = 1$, o que é equivalente a m ser um número primo. □

ANEXO

B

TABELA ASCII

A seguir apresentamos a tabela ASCII que contém alguns códigos binários.

Figura 8 – Tabela ASCII

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

Fonte: http://goo.gl/g0OpzN Acesso em 5 fev. 2016