Algebraic Coding Theory

Degree Thesis
BY
Brickner Ferenc Péter

BSc Mathematics

Mathematical Analyst Specialization

Supervisor: Kiss Emil
Head of Department
Department of Algebra and Number Theory
Eötvös Loránd University, Faculty of Science

Budapest
2019

Acknowledgement

I would like to very strongly emphasize the tremendous amount of help that I have been given by Kiss Emil. Without his conspicuous contribution this work could not have been finished that well. He was not only an excellent supervisor, but an outstanding mentor as well.

Additionally, everyone needs to be given credit who contributed to the obtainment of my capablilites throughout my journey. I am indeed enormously grateful for every bit of knowledge that I was capable of acquiring because of those individuals.

Contents

1 Introduction

The main idea behind this work is to capture the fundamental moments which were leading me towards acquiring a basic understanding of algebraic coding theory. I opted for this topic because of the fact that it is versatile in terms of possibilities for modern utilization. Furthermore I do have a personal emotional connection to this field of study.

Despite being a degree thesis, I am certainly hoping that people can use this work as a material for studying. Furthermore, I intentionally structured this work so that readers could utilize it as a source of learning. It means that for new concepts I have provided a plethora of examples, and a little bit longer intruduction.

This work is intended to present a short overview of Algebraic Coding Theory with the purpose of elaborating on a specific type of code, contemplating the possibility of its generalization, and then later briefly mentioning what kind of other codes exist. In the next section, the fundamental conceps concerning Algebraic Coding Theory will be elaborated on. These are easy to understand ideas, essential for developing a knowledge in this field of study. After that, an entire section dedicated only to Reed-Muller codes will follow. Different possible conceptualizations and definitions will be explored in depth. We will show the connection with Hadamard matrices and Conference matrices. We will explore the connection between geometry and Reed-Muller codes. The next section will provide an insight to a possible generalization of Reed-Muller codes. The last section of this work will demonstrate some of the other codes have been utilized by humanity in a brief manner.

2 Fundamental concepts, general discussion about codes

2.1 Fields, Matrices, Polynomials

The sources I have utilized to construct this subsection are [1],[2],[3], [4] and [6].

Since for the entirety of this work we will rely on the concept of fields, we briefly introduce it. The cited sources elaborate on these topics at length.

2.1.1 Fields

It will be obviously assumed that the reader is familiar with these fundamental concepts.

Definition Let

$$F = (K, 0, 1, +, *, \text{-}, ^{-1})$$

be an ordered 7-tuple where

$$0 \in K \wedge 1 \in K \wedge 0 \neq 1$$

furthermore

$$+ : K \times K \to K$$

and

$$* : K \times K \to K$$

are binary operations,

$$\text{-} : K \to K$$

and

$$^{-1} : K \setminus \{0\} \to K$$

are unary operations. The 7-tuplet F is called a **field** by definition if any only if the following criteria are met:

- The operations $+$ and $*$ are commutative, associative.
- The operation $*$ is distributive over operation $+$.
- For all $x \in K$ the element -x is the inverse of x with respect to operation $+$.
- For all $x \in K \setminus \{0\}$ the element x^{-1} is the inverse of x with respect to operation $*$.
- The 0 is the neutral element of operation $+$ and 1 is the neutral element of operation $*$.

The element 0 is said to be **additive identitiy** or **zero**, and the element 1 is the **multiplicative identity** or **one**. For $+$ and $*$ we will utilize the infix convention, for - the prefix notation will be used, and for $^{-1}$ we will use the postfix one. Obviously we can define **subtraction** and **division** over a field as well

$$- : K \times K \to K$$

$$/ : K \times K \setminus \{0\} \to K$$

by the following formulas

$$\forall (x, y) \in K \times K \ x - y = x + (\text{-}y)$$

$$\forall (x, y) \in K \times K \setminus \{0\} \ a/b = a * (b^{-1}).$$

Also the operations + and $*$ can be generalized so that they can have more than two or less than two variables. They are denoted by $\sum$ and $\prod$ respectively. What we have to consider while defining them is to pay attention to the 0-variable case, notably: and

$$\sum_{x\in\emptyset} x = 0$$

$$\prod_{x\in\emptyset} x = 1$$

If $K" \subseteq K$ is a system in K such that $\mathrm{Card}(K) \in \mathbb{N}^+$ then there exists a $\hat{x} \in K"$ element and the $\sum$ and $\prod$ operators are defined as

$$\sum_{x\in K"} x = \left[\sum_{x\in K"\setminus\{\hat{x}\}} x\right] + \hat{x}$$

$$\prod_{x\in K"} x = \left[\prod_{x\in K"\setminus\{\hat{x}\}} x\right] * \hat{x}.$$

Theorem 2.1 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field.*

- *The additive identity and the multiplicative identity are unique:*

$$\forall z \in K\ [\forall x \in K\ x + z = x] \to z = 0$$

$$\forall z \in K\ [\forall x \in K\ x * z = x] \to z = 1$$

- *The addivite inverse and the multiplicative inverse of an element are unique.*

$$\forall z \in K\ x + z = 0 \to z = \text{-}x$$

$$\forall z \in K\ x * z = 1 \to z = x^{-1}$$

- *The cancellative property holds.*

$$\forall x \in K\ \forall y \in K\ \forall z \in K\ x + y = x + z \leftrightarrow y = z$$

$$\forall x \in K \setminus \{0\}\ \forall y \in K\ \forall z \in K\ x * y = x * z \leftrightarrow y = z$$

- *For all $y \in K$ elements the $K \to K$ oprerations*

$$x \mapsto x + y$$

$$x \mapsto x - y$$

$$x \mapsto y - x$$

  are bijections.

- *For all $y \in K$ the $K \to K$ operation*

$$x \mapsto x * y$$

  is a bijection if and only if $y \in K \setminus \{0\}$.

- *For all $y \in K \setminus \{0\}$ the $K \to K$ operation*

  $$x \mapsto x/y$$

  is a bijection.

- *For all $y \in K$ the $K \setminus \{0\} \to K$ operation*

  $$x \mapsto y/x$$

  is a bijection if and only if $y \in K \setminus \{0\}$.

- *The $K \to K$ functions*

  $$x \mapsto x/1$$
  $$x \mapsto x - 0$$

  are the identity function.

- *Forall $x \in K$ we have $1/x = x^{-1}$.*

- *The elements $1 \in K$ and -1 $\in K$ are fixed points of the $K \setminus \{0\}$ operator*

  $$x \mapsto x^{-1}.$$

  (these two elements are not necesseraly distinct)

- *For all $x \in K \setminus \{0\}$*

  $$1/(1/x) = x$$
  $$(x^{-1})^{-1} = x$$
  $$x/x = 1$$

  hold.

- *For all $x \in K$ it holds that*

  $$\text{-}(\text{-}(x)) = x$$

  and

  $$x - x = 0.$$

- *For all $(x, y) \in K \times K$ the statement*

  $$\text{-}(x - y) = y - x$$

  is true.

- *For all $(x, y) \in (K \setminus \{0\}) \times (K \setminus \{0\})$ the equality*

  $$(x * y)^{-1} = x^{-1} * y^{-1}$$

  holds.

- *For all $(x, y) \in K \times (K \setminus \{0\})$ we have*

  $$y * (x/y) = x.$$

Proof All of these trivially follow from the definition of fields and the definition of subtraction and division. ■

Theorem 2.2 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field, and let $K” \subseteq K$ be such that Card$(K”) \in \mathbb{N}^+$. In this case*

$$\prod_{x \in K”} x = 0 \leftrightarrow \exists\ y \in K”\ \ y = 0.$$

Proof It is sufficient to show that the statement

$$\forall (a, b) \in K \times K\ \ a * b = 0 \leftrightarrow a = 0 \vee b = 0$$

holds, our theorem follows from it by induction. $\leftarrow$ Because 0 is the additive identity, we have $0 = 0 + 0$. Multiplying this equation by a we have that $a * 0 = a * (0 + 0)$ from which because of the distributivity of $*$ over $+$ we have $a * 0 = 0 + a * 0 = a * 0 + a * 0$. Because of the cancellative property of $+$ we obtain that $a * 0 = 0$.

$\rightarrow$ Let $a * b = 0$. If $a = 0$, then our proof is finished. If $a \neq 0$, then $a \in \text{Dom}(^{-1}) = K \setminus \{0\}$, and we obtain $a^{-1} * (a * b) = a^{-1} * 0$. The right side of the equation is zero because of $\leftarrow$ and the left side of the equation is b because of the associativity of $*$ and the fact that $a^{-1} * a = 1$ and 1 is the multiplicative identity, therefore $b = 0$.$\blacksquare$

2.1.2 Exponentiation over fields

We will see a lot of cases in this work when we will use exponentiation, for instance in5.1.2 therefore it is needed to mention them at least briefly. From now, $\mathbb{Z}$ symbol in the lower index will denote the following elements of and operations over $\mathbb{Z}$: addition, subtraction, multiplication, additive identity, multiplicative identity. The sum operator over $\mathbb{Z}$ will be denoted by $\sum^{(\mathbb{Z})}$.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. The

$$\hat{}: K \times \mathbb{N} \rightarrow K\ (x, n) \mapsto x^n$$

is defined in the following two steps:

$$\forall x \in K : x^{0_{\mathbb{Z}}} = 1$$

$$\forall x \in K\ \forall n \in \mathbb{N} \setminus \{0_{\mathbb{Z}}\} : x^n = x * x^{n -_{\mathbb{Z}} 1_{\mathbb{Z}}}.$$

The exponentiation can be generalized to negative exponents as well, but in this case $x \neq 0$.

$$\forall x \in K \setminus \{0\}\ \forall n \in \mathbb{Z} \setminus \mathbb{N} : x^n = 1/x^{\text{-}_{\mathbb{Z}} n}.$$

This operation along with the generalization is called **exponentiation** over the field F. The element $x \in K$ is called the **base**, the $n \in \mathbb{Z}$ element is called the **exponent**, and x^n is the **power**.

Theorem 2.3 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. The following are the exponential identities.*

- $\forall x \in K : x^{1_{\mathbb{Z}}} = x$
- $\forall n \in \mathbb{Z} : 1^n = 1$
- $\forall n \in \mathbb{N} \setminus \{0_{\mathbb{Z}}\} : 0^n = 0$
- *If $x \in K$ and $H \subset \mathbb{Z}$ are such that Card$(H) \in \mathbb{N}$ and for all $h \in H$ the power x^h is defined, then we have*

$$\prod_{h \in H} x^h = x^{\sum^{(\mathbb{Z})}_{h \in H} h}$$

- *If $K" \subseteq K$ and $n \in \mathbb{Z}$ are such that $Card(K") \in \mathbb{N}$ and x^n is defined for all $x \in K"$ then we have*
$$\left[\prod_{x \in K"} x\right]^n = \prod_{x \in K"} x^n.$$
- *If $(x^n)^m$ is defined, then $(x^n)^m = x^{n *_{\mathbb{Z}} m}$.*
- $\forall x \in K \setminus \{0\}$ *: $x^{\text{-}_{\mathbb{Z}} 1_{\mathbb{Z}}} = x^{-1} = 1/x$*

Proof All of them trivially follow from the definition. ■

2.1.3 Absolute value function over a field

Absolute value over a field is required to talk about normed spaces which is paramount when dealing with coding theory. Ineed linear codes 2.6.1 will form a normed space 2.2.8 with the weight function 2.5.2.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. Let $*_{\mathbb{R}}$ denote the mutliplication over $\mathbb{R}$ and $+_{\mathbb{R}}$ denote the addition over $\mathbb{R}$. Additionally let $\leq_{\mathbb{R}}$ be the ordering over $\mathbb{R}$. The

$$|.| : K \to \mathbb{R}$$

function is called an **absolute value function** over F by definition if and ony if the conditions

1. $\forall x \in K : |x| = 0_{\mathbb{R}} \leftrightarrow x = 0$ (uniqueness of the root)
2. $\forall (x, y) \in K \times K : |x * y| = |x| *_{\mathbb{R}} |y|$ (multiplicativity)
3. $\forall (x, y) \in K \times K : |x + y| \leq_{\mathbb{R}} |x| +_{\mathbb{R}} |y|$ (triangle inequality)
4. $\forall x \in K : |x| \geq_{\mathbb{R}} 0_{\mathbb{R}}$ (nonnegativity)

are met.

The trivial absolute value function shows us that for every field there is an absolute value function. It is defined by

$$|.|_{\text{triv}} : K \to \mathbb{R}$$
$$[\forall x \in K \ |x|_{\text{triv}} = 1_{\mathbb{R}} \leftrightarrow x \neq 0] \wedge |0|_{\text{triv}} = 0_{\mathbb{R}}.$$

By substituting $x = y = 1$ to the multiplicativity condition we obtain that

$$|1| = 1_{\mathbb{R}}$$

from which

$$\forall x \in K : |1/x| = (1_{\mathbb{R}})/_{\mathbb{R}}|x|$$

comes easily, and by substituting $x = y = -1$ we have

$$|-1| = 1_{\mathbb{R}}$$

from which one can show without difficulties that

$$\forall x \in K : |\text{-}x| = \text{-}_{\mathbb{R}}|x|$$

holds, where $1_{\mathbb{R}}$ is the multiplicative identity over the field of real numbers, and the $/_{\mathbb{R}}$ function is the division of real numbers, furthermore $-_{\mathbb{R}}$ is the operation which creates the additive inverse of a real number. From the multiplicativity of the absolute value function it can be proven via induction by Card(K") that if $K" \subseteq K$ is a system of elements such that Card(k") $\in \mathbb{N}$ we have

$$\left| \prod_{x \in K"} x \right| = \prod_{x \in K"}^{(\mathbb{R})} |x|$$

where $\prod^{(\mathbb{R})}$ is the multivariable multiplication over real numbers. Similarly it can be deduced from the triangle inequality that

$$\left| \sum_{x \in K"} x \right| \leq_{\mathbb{R}} \sum_{x \in K"}^{(\mathbb{R})} |x|$$

where $\sum^{\mathbb{R}}$ is the summation over real numbers. An example to be remembered from high school is the Euclidean absolute value

$$|.|_{\text{Eucl.}} : \mathbb{R} \to \mathbb{R}$$

defined by

$$\forall x \in \mathbb{R} : |x|_{\text{Eucl.}} = \text{Max}\{x, 0_{\mathbb{R}}\} + \text{Max}\{-x, 0_{\mathbb{R}}\} = \text{Max}\{x, 0_{\mathbb{R}}\} - \text{Min}\{x, 0_{\mathbb{R}}\}.$$

Another example which is not difficult to relate to is the complex absolute value function

$$|.| : \mathbb{C} \to \mathbb{R}$$

defined by

$$\forall z \in \mathbb{C} : |z| = \sqrt{\text{Re}(z) *_{\mathbb{R}} \text{Re}(z) +_{\mathbb{R}} \text{Im}(z) *_{\mathbb{R}} \text{Im}(z)}.$$

From the multiplicativity attribute it follows by induction that

$$\forall x \in K \ \forall n \in \mathbb{N} : |x^n| = |x|^n.$$

It can also be shown that for all $|.|$ absolute value functions we have

$$\forall (x, y) \in K \times K : ||x| -_{\mathbb{R}} |y||_{\text{Eucl.}} \leq_{\mathbb{R}} |x - y|.$$

This one is well-known from high school.

2.1.4 Matrices

In this work a plethora of different matrices will appear for a variety of specific purposes, therefore they need to have at least a short introduction.

Definition If S_1, S_2 and S_3 are sets, the

$$M : S_1 \times S_2 \to S_3$$

operations are called **matrices** over S_3.

The most obvious example for a matrix is the function

$$\emptyset \times \emptyset \to \emptyset.$$

The operation

$$\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$$

$$(x, y) \mapsto x +_{\mathbb{Z}} y$$

is a matrix over $\mathbb{Z}$. The binary function

$$\{0, 1\} \times \{0, 1\} \to \{0, 1\}$$

$$(a, b) \mapsto a \wedge b = \mathrm{Min}(a, b)$$

is a matrix over $\{0, 1\}$. Now we will only focus on matrices over fields, especially finite fields when dealing with coding theory.

Definition Let

$$F = (K, 0, 1, +, *, \text{-}, ^{-1})$$

be a field. Let

$$\mathbb{N}_{\leq n} = \mathbb{N} \cap [1, n].$$

The operation

$$M : \mathbb{N}_{\leq n} \times \mathbb{N}_{\leq k} \to K$$

is called a **matrix over the field F**. We utilize the notation

$$M(i, j) = M_{ij}.$$

We also use the notation

$$F^{n \times k} = \{M \mid M : \mathbb{N}_{\leq n} \times \mathbb{N}_{\leq k} \to K\}.$$

The elements of Image(M) are called the **entries** of a matrix. The elements of the sequence $(M_{ij})_{i \in \mathbb{N}_{\leq n}}$ are called the **rows** of the matrix and the elements of $(M_{ij})_{j \in \mathbb{N}_{\leq k}}$ are the **columns** of the matrix. The addition of the matrices happen entrywise. Let $M \in F^{n \times k}$ and $N \in F^{k \times s}$. Then the MN matrix is defined by its entries in the following way

$$(MN)_{ij} = \sum_{\gamma=1}^{k} M_{i\gamma} N_{\gamma j}.$$

The operation defined in this way is called **matrix multuplication** and is a

$$F^{n \times k} \times F^{k \times s} \to F^{n \times s}$$

binary operation. The **multiplication of matrices by a scalar** is defined entrywise. In the case of $n = k$ the matrix is called a **square matrix**.

Obviously the multiplication of matrices is associative but not commutative. The addition of matrices is both associative and commutative. The distributivity attribute holds as well both for matrix multiplication and scalar multiplication. The additive identity is the zero matrix, whose entries are all zeros. As an example, look at the matrix E_n defined by its entries as

$$E_{n_{ij}} = 0 \leftrightarrow i \neq j \wedge E_{n_{ij}} = 1 \leftrightarrow i = j.$$

It is clear that

$$\forall M \in F^{n \times n} : ME_n = E_n M = M.$$

We will now look at the concept of the transpose matrix. It will appear in this work later on, for example in 2.26.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. The ^{F} : $F^{n\times k} \to F^{k\times n}$ function is defined as

$$\forall M \in F^{n\times k} : \forall (i,j) \in \mathbb{N}_{\leq n} \times \mathbb{N}_{\leq k} : M^F_{ij} = M_{ji} \in F^{k\times n}$$

The matrix $M^T \in F^{k\times n}$ is said to be by definition the **transpose** of the matrix M.

Now we will formulate some of the most elementary properties of transpose matrices.

Theorem 2.4 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field.*

- $\forall M \in F^{k\times n}$ *:* $(M^T)^T = M$
- *Let* $\overrightarrow{\prod}$ *denote the the multivariable matrix multiplication now. Let* $(M_\gamma)_{\gamma=1}^t$ *a finite sequence of matrices where the multiplication is defined, furthermore* $t \geq 1$*. The statement*

  $$\left[\overrightarrow{\prod_{\gamma=1}^{\vec{t}}} M_\gamma\right]^T = \overrightarrow{\prod_{\gamma=1}^{\vec{t}}} M^T_{t+1-\gamma}$$

  holds.
- $$\forall (M,N) \in T^{n\times k} \times T^{k\times s} : (MN)^T = N^T M^T$$

  (this is the $t = 2$ special case of the previous one)
- *Let* $\overrightarrow{\sum}$ *denote the multivariable matrix addition now. Let* M” *a finite set of matrices with the same size, additionally* $Card(M”) \geq 1$*. Then we have*

  $$\left[\overrightarrow{\sum_{M\in M”}} M\right]^T = \overrightarrow{\sum_{M\in M”}} M^T$$
- *Let* $\vec{+}$ *denote the addition of matrices now. The same notation will be utilized for vector spaces in general concerning the first part of the first section. We have*

  $$\forall (M,N) \in F^{n\times k} \times F^{k\times s} : (M \vec{+} N)^T = M^T \vec{+} N^T.$$

  (this is the $t = 2$ special case of the previous one)
- $\forall \lambda \in K\ \forall M \in F^{n\times k}$ *:* $(\lambda M)^T = \lambda M^T$
- $\forall n \in \mathbb{N} \setminus \{0\}$ *:* $E_n^T = E_n$.

2.1.5 Determinant

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. Let $P(\mathbb{N}_{\leq n})$ denote the set of all permutations on $\mathbb{N}_{\leq n}$. Consider the operation

$$\text{Det} : F^{n\times n} \to K$$

defined by the formula

$$\text{Det}(M) = \sum_{\sigma\in P(\mathbb{N}_{\leq n})} (\text{-}1)^{f(\sigma)} \left[\prod_{i\in\mathbb{N}_{\leq n}} M_{i\sigma(i)}\right]$$

where $f : P(\mathbb{N}_{\leq n}) \to \mathbb{N}$ is the function which tells us how many inversions does the permutation have. The

$$\text{Det} M \in K$$

element is the **determinant** of the matrix M.

Theorem 2.5 *Let $\lambda \circ^{\gamma} M$ mean that the γ-th row of M is multiplied by λ. For all $\lambda \in K$ and for all γ indices, furthermore for all M matrix we have*

$$Det(\lambda \circ^{\gamma} M) = \lambda * Det(M).$$

Theorem 2.6 *Let the multivariable matrix multiplication be denoted by $\overset{\rightarrow}{\prod}$ If M" is a system of matrices of same size and $Card(M) \geq 1$ the statement*

$$Det\left(\overset{\rightarrow}{\prod_{M \in M"}} M \right) = \prod_{M \in M"} Det(M)$$

holds.

We note that the theorem works for the empty system of matrices as well if the empty products are defined so that $\overset{\rightarrow}{\prod}_{M \in \emptyset} M = E_n$ and $\prod_{x \in \emptyset} = 1$ It can also be shown that

$$\forall M \in F^{n \times n} : \mathrm{Det}(M) = \mathrm{Det}(M^T).$$

2.1.6 Vandermonde matrices and Vandermonde determinants

Vandermonde matrices and determinans are mentioned because of their presence in this work, for instance in 2.8.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ a fiield, and let

$$K" = (\lambda_1, \lambda_2, \ldots \lambda_n) \subseteq K$$

be a sequence of elements. The matrix

$$\text{Vandermonde}(K") = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & \ldots & \lambda_1^{n-1} \\ 1 & \lambda_2 & \lambda_2^2 & \ldots & \lambda_2^{n-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & \lambda_n & \lambda_n^2 & \ldots & \lambda_n^{n-1} \end{pmatrix}$$

is called a **Vandermonde matrix**. The determinant of the Vandermonde matrix is called **Vandermonde determinant**. The elements

$$\lambda_1, \lambda_2, \ldots \lambda_n$$

are called the **generators** of the Vandermonde matrix/determinant.

It is not difficult to show that

$$\mathrm{Det}(\text{Vandermonde}(K")) = \prod_{1 \leq \gamma_1 < \gamma_2 \leq n} [\lambda_{\gamma_1} - \lambda_{\gamma_2}]$$

holds.

Theorem 2.7 *The Vandermonde determinant is zero if and only if there are two generators with the same value.*

Proof Because of 2.2 the value of

$$\prod_{1 \leq \gamma_1 < \gamma_2 \leq n} [\lambda_{\gamma_1} - \lambda_{\gamma_2}]$$

is zero if and only if there is an (γ_1, γ_2) pair of indices so that $\gamma_1 \neq \gamma_2$ and $\lambda_{\gamma_1} - \lambda_{\gamma_2} = 0$, which is equivalent with the statement that there are generators λ_{γ_1} and λ_{γ_2} such that $\lambda_{\gamma_1} = \lambda_{\gamma_2}$, proving the statement. ■

The type of matrix we will have in 2.8 has the following form. Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $K" = (\lambda_1, \lambda_2, \ldots \lambda_n) \subseteq K$ be a system of elements. Furthermore let

$$\Psi = (\Psi_1, \Psi_2, \ldots \Psi_n) \subseteq K$$

be a sequence of elements as well. Then our matrice is defined as

$$M(\Psi, \text{Vandermonde}(K")) = \begin{pmatrix} \Psi_1 & \Psi_1 * \lambda_1 & \Psi_1 * \lambda_1^2 & \ldots & \Psi_1 * \lambda_1^{n-1} \\ \Psi_2 & \Psi_2 * \lambda_2 & \Psi_2 * \lambda_2^2 & \ldots & \Psi_2 * \lambda_2^{n-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \Psi_n & \Psi_n * \lambda_n & \Psi_n * \lambda_n^2 & \ldots & \Psi_n * \lambda_n^{n-1} \end{pmatrix}.$$

Theorem 2.8

$$\mathit{Det}(M(\Psi, \mathit{Vandermonde}(K"))) = \prod_{\gamma=1}^{n} \Psi_\gamma \prod_{1 \leq \gamma_1 < \gamma_2 \leq n} [\lambda_{\gamma_1} - \lambda_{\gamma_2}]$$

Proof The matrix $M(\Psi, \text{Vandermonde}(K"))$ is obtained from Vandermonde$(K")$ by multiplying the γ-th row by Ψ_γ. Using 2.5 and the explicit formula for the Vandermonde determinant the statement follows immediately. ■

Theorem 2.9

$$\mathit{Det}(M(\Psi, \mathit{Vandermonde}(K"))) = 0 \leftrightarrow \exists \gamma \ \Psi_\gamma = 0 \vee \exists (\gamma_1, \gamma_2) \ \lambda_{\gamma_1} = \lambda_{\gamma_2} \wedge \gamma_1 \neq \gamma_2$$

Proof The statement follows easily from 2.2 and 2.8. ■

2.1.7 Polynomials over a field

A lot of polinomials will appear when elaborating on codes, therefore it is obligatory to mention them. Note that there is a difference between polynomials and polynomial functions, but making a distinction is not needed for our purposes.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. Let $S(K)$ denote the set of all sequences over K. The set

$$F[X] = \{s \mid s \in S(K) \wedge \text{Card}\{s' \in s \mid s' \neq 0\} \in \mathbb{N}\}$$

is called the set of **polynomials** over the field F. An $f \in F[X]$ is called a **polynomial** over F. The elements of the sequence are called the **coefficients** of the polynomial. The polinomial whose coefficients are all 0 is called the **zero polynomial** and is now denoted by $f = \vec{0}$.

In other words the polynomials over a field are exactly those sequences which contain only a finite number of nonzero elements.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $f \in F[X]$ be a polynomial. If the

$$\text{Max}\{i \in \mathbb{N} \mid a_i \neq 0\} \in \mathbb{N}$$

element exists, then we call it the **degree of the polynomial** f and we write

$$\text{Deg}(f) = \text{Max}\{i \in \mathbb{N} \mid a_i \neq 0\} \in \mathbb{N}.$$

We can easily observe that aside from the polynomial $f = \vec{0}$ all polynomials have an unique degree, meaning that

$$\text{Deg} : F[X] \setminus \{\vec{0}\} \to \mathbb{N}$$

is a well-defined function. The nonzero

$$f = (f_i)_{i \in \mathbb{N}}$$

polynomial is often depicted by the unary operation

$$K \to K$$

$$f(X) = \sum_{\gamma=0}^{\text{Deg}(f)} f_i X^i.$$

We can also evaluate the polynomial for a $x \in K$ element

$$f(x) = \sum_{\gamma=0}^{\text{Deg}(f)} f_i x^i \in K.$$

The addition and multiplication by scalar is defined coefficientwise. Multiplication of polynomials comes naturally as well. Obviously the addition of polynomials is commutative and associative, the multiplication by scalar and multiplication of polynomials are distributive over the operation of the addition of polynomials.Obviously the polynomial $\vec{0}$ is the neutral element of the addition of polynomials, and the $1 \in K$ scalar is the neutral element of scalar multiplication. The polynomial represented by the function $f(X) = X^0$ is the neutral element of the multiplication of polynomials.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. Let $f(X) \in F[X]$ be a polynomial over F. The $x \in K$ element is said to be the **root** of the polynomial by definition if and only if $f(x) = 0$. A polynomial is said to be **monic** by definition if and only if

$$f_{\text{Deg}(f)} = 1.$$

2.1.8 The fundamental theorem of algebra

The following theorem is not only paramount when we are trying to comprehend algebra in general, but we will use it in this work for example in 5.1.2. This theorem is known as the **fundamental theorem of algebra**.

Theorem 2.10 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $f(X) \in F[X]$ a monic polynomial. Let the multivariable multiplication of polynomials be denoted by $\vec{\prod}$. Then*

$$Card\{x \in K \mid f(x) = 0\} \leq Deg(f)$$

holds. In other words, a polynomial over a field cannot have more roots than its degree. Furthermore if

$$Card\{x \in K \mid f(x) = 0\} = Deg(f)$$

then we have

$$f(X) = \overrightarrow{\prod}_{x \in K : f(x)=0} (X - x).$$

The example $f(X) = X^2 + 1$ over $\mathbb{C}$ is easy to relate to. Indeed, the roots of f are $\pm i$ and

$$f(X) = X^2 + 1 = (X - i)(X + i)$$

The theorem cannot necessarily be generalized to polynomials over any structure. For example the polynomial $f(X) = X^2$ over the ring of dual numbers have a degree of 2, but has infinite roots, for example

$$f(a *_{\mathbb{D}} \epsilon) = (a *_{\mathbb{D}} \epsilon)^2 = a^2 *_{\mathbb{D}} \epsilon^2 = a^2 *_{\mathbb{D}} 0_{\mathbb{D}} = 0_{\mathbb{D}}$$

for all $a \in \mathbb{R}$ real number.

2.2 Vector Spaces

For this subsection I utilized the following sources: [2], [1], [5].

2.2.1 Definition and elementary properties

Definition Let

$$F = (K, 0, 1, +, *, \text{-}, ^{-1})$$

be a field. Let

$$W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$$

be a 6-tuple, where V is a set

$$\vec{0} \in V$$

is an element of the set and

$$\vec{+} : V \times V \to V$$

and

$$\vec{*} : K \times V \to V$$

are a binary operatios and

$$\vec{-} : V \to V$$

is a unary operation. The 6-tuple W is called a **vector space** by definition if and only if the following criteria are met.

- The operation $\vec{+}$ is commutative and associative.
- The element $\vec{0} \in V$ is the identity element of the operation $\vec{+}$.
- The element $1 \in K$ is the left identity element of the operation $\vec{*}$.
- For all $v \in V$ the element $\vec{-}v$ is the inverse of v with respect to operation $\vec{+}$. (additive inverse)

- For all
$$(\lambda_1, \lambda_2, v) \in K \times K \times V$$
triplets we have
$$(\lambda_1 + \lambda_2)\vec{*}v = \lambda_1\vec{*}v\vec{+}\lambda_2\vec{*}v$$
and
$$\lambda_1\vec{*}(\lambda_2\vec{*}v) = (\lambda_1 * \lambda_2)\vec{*}v.$$

- For all
$$(\lambda, v_1, v_2) \in K \times V \times V$$
triplets we have
$$\lambda\vec{*}(v_1\vec{+}v_2) = \lambda\vec{*}v_1\vec{+}\lambda\vec{*}v_2.$$

The elements of K are called **scalars** and the elements of V are called **vectors** The element $\vec{0}$ is called the **zero vector**. the operation $\vec{+}$ is called the **addition of vectors** and the binary operation $\vec{*}$ is called the **scalar multiplication of vectors**. Of course we can define **subtraction** of vectors as well
$$\vec{-} : V \times V \to V$$
by the formula
$$\forall (v_1, v_2) \in V \times V : v_1\vec{-}v_2 = v_1\vec{+}(\vec{\text{-}}v_2)$$
We can also say that W is a vector space over field F. Sometimes the notation is W_F. Obviously the function $\vec{+}$ can be generalized to more or less variables and is denoted by $\vec{\sum}$ with te empty sum being
$$\vec{\sum_{x\in\emptyset}} x = \vec{0}$$
and if $V” \subseteq V$ is a system of vectors such that $\text{Card}(V”) \in \mathbb{N}^+$ then there is a $\widehat{v} \in V”$ vector and our definition is
$$\vec{\sum_{v\in V”}} v = \left[\vec{\sum_{v\in V”\setminus\{\widehat{v}\}}} v\right]\vec{+}\widehat{v}.$$

As an easy example if $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ is a field, then
$$W = (K, F, 0, +, *, \text{-})$$
is a vector space. Now we will examine some of the fundamental properties of vector spaces.

Theorem 2.11 *Let $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{\text{-}})$ be a vector space over the field $F = (K, 0, 1, +, *, \text{-}, ^{-1})$. The following hold.*

- *The additive identity is unique, meaning that*
$$\forall v_1 \in V : [\forall v_2 \in V \ v_1\vec{+}v_2 = v_2] \to v_1 = \vec{0}.$$

- *The lef identity of $\vec{*}$ is unique, meaning that*
$$\forall \lambda \in K : [\forall v \in V \ \lambda\vec{*}v = v] \to \lambda = 1.$$

- *For all* $v_2 \in V$ *the functions*
$$v_1 \mapsto v_1 \vec{+} v_2$$
and
$$v_1 \mapsto v_1 \vec{-} v_2$$
and
$$v_1 \mapsto v_2 \vec{-} v_1$$
are bijective.

- *For all* $\lambda \in K$ *the function*
$$v \mapsto \lambda \vec{*} v$$
is a bijection if and only if $\lambda \in K \setminus \{0\}$.

- *For all* $v \in V$ *the function*
$$\lambda \mapsto \lambda \vec{*} v$$
is a bijection if and only if $v \neq \vec{0}$.

- *For all* $(v_1, v_2) \in V \times V$ *we have*
$$\vec{-}(v_1 \vec{-} v_2) = v_2 \vec{-} v_1.$$

- *For all* $v \in V$ *we have* $\vec{-}v = (\text{-}1) \vec{*} v$.

Proof All of these follow immediately from the definition of vector spaces and subtraction and the fundamental properties of fields. ■

Theorem 2.12 *Let* $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ *be a vector space over the field* $F = (K, 0, 1, +, *, \text{-}, ^{-1})$. *For all*
$$(\lambda, v) \in K \times V$$
we have
$$\lambda \vec{*} v = \vec{0} \leftrightarrow \lambda = 0 \vee v = \vec{0}.$$

Proof Indeed
$$0 \vec{*} v = (0 + 0) \vec{*} v = 0 \vec{*} v \vec{+} 0 \vec{*} v$$
from where because of the uniqueness of additive identity it is obtained that $0 \vec{*} v = \vec{0}$. Now it is easy to be deduced that
$$\lambda \vec{*} \vec{0} = \lambda \vec{*} (\vec{0} \vec{+} \vec{0}) = \lambda \vec{*} \vec{0} \vec{+} \lambda \vec{*} \vec{0}$$
from which because of the very same reason we get $\lambda \vec{*} \vec{0} = \vec{0}$. Now let $\lambda \vec{*} v = \vec{0}$. If $\lambda = 0$, then we have arrived to our conclusion. If $\lambda \neq 0$ then $\lambda \in \mathrm{Dom}(^{-1})$, consequently
$$v = 1 \vec{*} v = (\lambda^{-1} * \lambda) \vec{*} v = \lambda^{-1} \vec{*} (\lambda \vec{*} v) = \lambda^{-1} \vec{*} 0 = \vec{0}$$
finishing our proof.■

2.2.2 Subspaces

Now we will briefly mention the concept of subspaces. Since linear codes 2.6.1 will be subspaces of a vector field, they require at least a short indroduction.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{\text{-}})$ a vector space. If

$$\widehat{W} = (\widehat{V}, F, \vec{0}, \widehat{\vec{+}}, \widehat{\vec{*}}, \widehat{\vec{\text{-}}})$$

is also a vector space and $\widehat{V} \subseteq V$ furthermore $\widehat{\vec{+}}$ is the restriction of $+$ to the set $\widehat{V} \times \widehat{V}$ and $\widehat{\vec{*}}$ is the restriction of $\vec{*}$ to the domain $K \times \widehat{V}$ additionally $\widehat{\vec{\text{-}}}$ is the restriction of $\vec{\text{-}}$ to the domain $\widehat{\vec{V}}$. In this case we say that $\widehat{W}$ is the **subspace** or **linear subspace** of W and we write $\widehat{W} \leq W$.

Obviously every vector space is a subspace of itself $W \leq W$ meaning that $\leq$ is a reflexive relation. The relation is also transitive

$$\widehat{\widehat{W}} \leq \widehat{W} \wedge \widehat{W} \leq W \to \widehat{\widehat{W}} \leq W.$$

One can easily see that the relation is antisymmetric

$$\widehat{W} \leq W \wedge W \leq \widehat{W} \to W = \widehat{W}.$$

We can observe without difficulties that

$$(\{0\}, F, \vec{0}, \vec{+}, \vec{*}, \vec{\text{-}}) \leq W$$

holds as well. Furthermore if

$$\widehat{\widehat{W}} \leq \widehat{W} \leq W \wedge \widehat{\widehat{W}} = W \to \widehat{\widehat{W}} = \widehat{W} = W.$$

Indeed, these properties are all consequences of the fundamental properties of $\subseteq$ relation. A very easy to understand example would be

$$\mathbb{Q}_{\mathbb{Q}} \leq \mathbb{R}_{\mathbb{Q}} \leq \mathbb{C}_{\mathbb{Q}}.$$

2.2.3 Generating system, basis, dimension

Now we will examine briefly the concept of dimension, which is paramount when dealing with coding theory.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{\text{-}})$ a vector space. The $B \subseteq V$ det is called a **basis set** of W by definition if and only if for every $v \in V$ there is exactly one

$$\lambda_v : B \to K$$

function that

$$v = \vec{\sum_{b \in B}} \lambda_v(b) \vec{*} b$$

holds. The $G \subseteq V$ set is called a **generating set** of W by definition if and only if for every $v \in V$ there is at least one

$$\lambda_v : B \to K$$

function that satisfies

$$v = \vec{\sum_{g \in G}} \lambda_v(g) \vec{*} g.$$

The $B = (b_i)_{i=1}^n \subseteq V$ system is called a **basis** of W by definition if and only if for every $v \in V$ there exists exactly one $\lambda_{i=1}^n \subseteq K$ system which satisfies the condition

$$v = \sum_{i=1}^{\vec{n}} \lambda_i \vec{*} b_i.$$

The $B = (b_i)_{i=1}^n \subseteq V$ system is called a **generating system** of W by definition if and only if for every $v \in V$ there exists at least one $\lambda_{i=1}^n \subseteq K$ system which satisfies the condition

$$v = \sum_{i=1}^{\vec{n}} \lambda_i \vec{*} b_i.$$

One can prove that for every generating system G there is a $G' \subseteq G$ system such that $G \setminus G'$ is a basis. It can be shown as well that if B_1 is a basis of F and B_2 is a basis of F as well then

$$\mathrm{Card}(B_1) = \mathrm{Card}(B_2)$$

holds. It can also be shown pretty easily that every vector space has at least one basis, so the concept of dimension is well-defined.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ a vector space and let $B(W)$ be a basis of W. The **dimension** of a vector space is the number of elements contained by $B(W)$. We use the notation

$$\mathrm{Dim}(W) = \mathrm{Card}(B(W)).$$

From now we will only discuss vector spaces where $\mathrm{Dim}(W) \in \mathbb{N}$. It can be seen fairly effortlessly that

$$\widehat{W} \leq W \to \mathrm{Dim}(\widehat{W}) \leq_{\mathbb{N}} \mathrm{Dim}(W)$$

It is easily observed that matrices of a given size form a vector space over a field. Furthermore

$$\widehat{W} \leq W \wedge \mathrm{Dim}(\widehat{W}) = \mathrm{Dim}(W) \to \widehat{W} = W$$

holds as well. A plethora of elementary consequences can be deduced merely by looking at the above definitions. For example it can be seen that if B is a basis, then $\vec{0} \notin B$.

2.2.4 Linear independence and rank

Linear independence is an insurmountable-to-avoid concept when dealing with coding theory. It appears in a plethora of different contexts in this work, for example in 2.26.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ a vector space. The $V" \subseteq V$ subset is called a **set of linearly independent vectors** by definition if and only if for every

$$\lambda : V" \to K$$

function if

$$\vec{\sum}_{v \in V"} \lambda(v) \vec{*} v = \vec{0}$$

then we have

$$\lambda \equiv 0.$$

The $V” = (v_i)_{i=1}^n \subseteq V$ system of vectors is called a **system of linearly independent vectors** if and only if for every $(\lambda_i)_{i=1}^n \subseteq K$ sequence of scalars we have

$$\sum_{i=1}^{\vec{n}} \lambda_i \vec{*} v_i = \vec{0} \to \forall i : \lambda_i = 0.$$

It can be easily shown that if $V” \subseteq V$ is a linearly independent system and

$$\mathrm{Card}(V”) = \mathrm{Dim}(W) \in \mathbb{N}$$

then $V”$ is a basis of W. Another elementary attribute of linearly independent systems is that they cannot contain the zero vector. Given a linearly independent system $V”$ for all $v \in V”$ the system $V” \setminus \{v\}$ is also a linearly independent set. A specific case appears if the rows or columns of a matrix over a field are regarded as vectors of a vector space.

Definition The **rank** of a matrix is the maximum number of linearly independent rows that can be selected from its rows.

We note that the very same approach works with columns as well and gives us entirely the same concept.

Definition The $n \times n$ matrix Ni s said to be **invertible** by definition if and only if there exists an M^{-1} matrix so that

$$MM^{-1} = M^{-1}M = E_n$$

holds. The matrix M^{-1} is by definition called the **inverse** of the matrix M.

The matrix M is invertible if and only if $\mathrm{Det}(M) \neq 0$ which is equivalent with the condition that the rows/columns of the matrix are linearly independent.

2.2.5 Linear maps

Linear maps and bilinear maps occur everywhere in coding theory in a wide range of contexts, therefore we will have a closer look on them. Furthermore the image and kernel of a linear map are paramount concepts to mention as well, since we will encounter them a lot, for instance in 2.6.1.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and

$$W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$$

$$\widehat{W} = (\widehat{V}, F, \widehat{\vec{0}}, \widehat{\vec{+}}, \widehat{\vec{*}}, \widehat{\vec{-}})$$

two vector spaces. The function

$$\mathrm{Lin} : V \to \widehat{V}$$

is called a **linear map** by definition if and only if the following two conditions are satisfied:

$$\forall (\lambda, v) \in K \times V : \mathrm{Lin}(\lambda \vec{*} v) = \lambda \widehat{\vec{*}}\, \mathrm{Lin}(v)$$

$$\forall (v_1, v_2) \in V \times V : \mathrm{Lin}(v_1 \vec{+} v_2) = \mathrm{Lin}(v_1) \widehat{\vec{+}} \mathrm{Lin}(v_2).$$

The **kernel** of a linear map is the set defined by

$$\mathrm{Ker}(\mathrm{Lin}) = \{v \in V \mid \mathrm{Lin}(v) = \widehat{\vec{0}}\} \subseteq V.$$

In other words the kernel is the set of those vectors which are sent to the zero vector of the second vector space by the linear function. The **image** of a linear map is the set defined by

$$\mathrm{Im}(\mathrm{Lin}) = \{\mathrm{Lin}(v) \mid v \in V\} = \{u \in \widehat{V} \mid \exists v \in V : \mathrm{Lin}(v) = u\} \subseteq \widehat{V}.$$

Two linear maps can be added pointwise and multiplied with a scalar pointwise.

It can be shown by induction that if $(v_i)_{i\in I}$ is a finite system of vectors and $(\lambda_i)_{i\in I}$ is a finite system of scalars then

$$\mathrm{Lin}\left(\vec{\sum_{i\in I}}\lambda_i \vec{*} v_i\right) = \sum_{i\in I} \lambda_i \widehat{\vec{*}}\, \mathrm{Lin}(v_i).$$

We can easily see that linear maps for the pointwise addition and pointwise multiplication by scalar along with the zero map form a vector space. It is usually denoted by

$$\mathrm{Hom}(W, \widehat{W}).$$

We can also observe without difficulties that

$$\mathrm{Ker}(\mathrm{Lin}) \subseteq V$$

forms a subspace of W. Now we will denote this subspace by

$$\mathrm{KER}(\mathrm{Lin}) \leq W.$$

It can be deduced fairly easily that

$$\mathrm{Im}(\mathrm{Lin}) \subseteq \widehat{V}$$

forms a subspace of $\widehat{W}$. This subspace will be denoted now by

$$\mathrm{IM}(\mathrm{Lin}) \leq \widehat{W}.$$

It can be shown that

$$\mathrm{Dim}(\mathrm{KER}(\mathrm{Lin})) + \mathrm{Dim}(\mathrm{IM}(\mathrm{Lin})) = \mathrm{Dim}(W).$$

It can be deduced effortlessly that

$$\mathrm{Lin}(\vec{0}) = \widehat{\vec{0}}.$$

It comes naturally from the first condition and from the elementary properties of vector spaces that

$$\mathrm{Lin}(\vec{0}) = \mathrm{Lin}(0 \vec{*} \vec{0}) = 0 \widehat{\vec{*}}\, \mathrm{Lin}(\vec{0}) = \widehat{\vec{0}}.$$

The same fact can be seen from the second condition and from the fact that only the zero vector is the additive identity in a vector space, so we have

$$\mathrm{Lin}(\vec{0} \vec{+} \vec{0}) = \mathrm{Lin}(\vec{0}) \widehat{\vec{+}} \mathrm{Lin}(\vec{0}) \to \mathrm{Lin}(\vec{0}) = \mathrm{Lin}(\vec{0}) \widehat{\vec{+}} \mathrm{Lin}(\vec{0}) \to \mathrm{Lin}(\vec{0}) = \widehat{\vec{0}}.$$

We can also obtain that

$$\forall v \in V : \mathrm{Lin}(\vec{-} v) = \widehat{\vec{-}}\, \mathrm{Lin}(v)$$

and additionally that

$$\forall (v_1, v_2) \in V \times V : \mathrm{Lin}(v_1 \vec{-} v_2) = \mathrm{Lin}(v_1) \widehat{\vec{-}} \mathrm{Lin}(v_2).$$

2.2.6 Linear span

Now we will continue our brief introduction to vector spaces with the concept of linear span. They will appear in this work as well, for instance in 5.3 and obviously everywhere in linear algebra.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and let $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ be a vector space. Let $V" \subseteq V$ be a system of vectors. The

$$\mathrm{LC}(V") = \left\{ \sum_{\forall v \in V"}^{\rightarrow} \lambda(v) \vec{*} v \mid \forall\ \lambda : V" \to K \right\} \subseteq V$$

set contains the **linear combinations** of the elements of $V"$. This forms a vector space with the restriced operations, which is usually denoted by $\mathrm{Span}(V")$ or by writing the elements of $V"$ between the symbols $<$ and $>$. The subspace is often referred to as the **linear span** of $V"$. Formally, if the restricted operations are denoted with the symbol $\cap$ in the lower index, then

$$\mathrm{Span}(V") = (\mathrm{LC}(V"), F, \vec{0}, \vec{+}_{\cap}, \vec{*}_{\cap}, \vec{-}_{\cap}) \leq W.$$

From the definition of a basis and a generating system it is clear that if B is a basis and G is a generating system of W, then

$$\mathrm{Span}(B) = \mathrm{Span}(G) = \mathrm{Span}(V) = W$$

Of course

$$V" \subseteq V"_2 \to \mathrm{Span}(V") \leq \mathrm{Span}(V"_2)$$

holds. For example if W is the vector space of planar vectors, then $< (0,1), (1,0) >$ will be the entire vector space, and $< (1,1) >$ is the subspace of those vectors whose two coordinates are the same, and $< (0,0) >$ is the vector space whose only vector is $\vec{0}$. The subspace $< (1,1), (2,2) >$ is not the entire space, because $(1,1)$ and $(2,2)$ are not linearly independent. Furthermore it is not difficult to observe that

$$\mathrm{LC}(V") = \bigcap_{V" \subseteq J \subseteq V \wedge \text{'J forms a vector space with the restricted operations'}} J$$

which means that the linear combinations of these vectors can be obtained as an intersection of all those subsets which contain the aforementioned vectors and forms a subspace with the operarations restricted.

2.2.7 Bilinear maps, scalar product

Bilinear maps will appear for instance in 2.6.3 and in 3.4.1, therefore they deserve a short discussion.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and let

$$W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$$

$$\widehat{W} = (\widehat{V}, F, \widehat{\vec{0}}, \widehat{\vec{+}}, \widehat{\vec{*}}, \widehat{\vec{-}})$$

$$\widehat{\widehat{W}} = (\widehat{\widehat{V}}, F, \widehat{\widehat{\vec{0}}}, \widehat{\widehat{\vec{+}}}, \widehat{\widehat{\vec{*}}}, \widehat{\widehat{\vec{-}}})$$

be three vector spaces. The function

$$\mathrm{BLN} : V \times \widehat{V} \to \widehat{\widehat{V}}$$

is called a **bilinear map** by definition if the following criteria are met.

- For all $v \in V$ the $u \mapsto \mathrm{BLN}(v,u)$ univariable function is a linear map.
- For all $v \in \widehat{V}$ the $u \mapsto \mathrm{BLN}(u,v)$ univariable function is a linear map.

In other words a bilinear map is a function with two variables which when restricted to any of its variables then becomes a linear map. The scalar product is a special example we will see along with the Hadamard product. The Hadamard product will be elaborated on in 3.4.1.

Definition $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and let $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ be a vector space. Of course $W = (K, F, 0, +, *, \text{-})$ is a vector space as well, because every field is a vector space over itself. The

$$\langle , \rangle : V \times V \to K$$

function defined by the foormula

$$\forall (v,w) \in V \times V : \langle v, w \rangle = \sum_{\forall i \in \mathbb{N}_{\leq \mathrm{Dim}(W)}} \mathrm{coord}_i(v) * \mathrm{coord}_i(w)$$

is called **scalar product**. The v and w vectors are said to be orthogonal by definition if and only if $\langle v, w \rangle = 0$ and is denoted by $v \perp w$.

Obviously the scalar product is a bilinear map. Furthermore because of the commutativity of $*$ we get

$$\forall (v,w) \in V \times V : \langle v, w \rangle = \sum_{\forall i \in \mathbb{N}_{\leq \mathrm{Dim}(W)}} \mathrm{coord}_i(v) * \mathrm{coord}_i(w) =$$

$$= \sum_{\forall i \in \mathbb{N}_{\leq \mathrm{Dim}(W)}} \mathrm{coord}_i(w) * \mathrm{coord}_i(v) = \langle w, v \rangle.$$

2.2.8 Norms and normed vector spaces

Norms will be used a lot when elaborating on coding theory.

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ be a vector space. Let the $|.| : K \to \mathbb{R}$ function be an absolute value function over F. The

$$\|.\| : V \to \mathbb{R}$$

function is a **norm** on W by definition if and only if the following criteria are met:

- $\forall v \in V : \|v\| \geq_{\mathbb{R}} 0_{\mathbb{R}}$ (nonnegativity)
- $\forall \lambda \in K \; \forall v \in V : \|\lambda \vec{*} v\| = |\lambda| *_{\mathbb{R}} \|v\|$ (homogenity)
- $\forall v_1 \in V \; \forall v_2 \in V : \|v_1 \vec{+} v_2\| \leq_{\mathbb{R}} \|v_1\| +_{\mathbb{R}} \|v_2\|$ (triange inequality)
- $\forall v \in V : \|v\| = 0_{\mathbb{R}} \leftrightarrow v = \vec{0}$ (uniqueness of the root)

We say that $\|.\|$ is **homogeneous** with respect to $|.|$. The ordered pair

$$(W, \|.\|)$$

is called in this case a **normed vector space**.

For instance the field $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ is a vector space over itself $W = (K, F, 0, +, *, \text{-})$, so a $|.| : K \to \mathbb{R}$ absolute value function will be a norm as well. From this example it can be observed that the concept of norm is a generalization of the concept of absolute value function. Our other example is located in 2.5. It is not difficult to deduce that every norm is homogeneous with respect to exactly one absolute value function for $\text{Card}(V) \neq 1$, since from

$$\forall \lambda \in K \ \forall v \in V : \|\lambda * v\| = |\lambda|_1 *_{\mathbb{R}} \|v\| = |\lambda|_2 *_{\mathbb{R}} \|v\|$$

substituting $v \neq \vec{0}$ we have

$$|.|_1 \equiv |.|_2.$$

From the triangle inequality we can obtain the generalized triangle inequality. For the $V" \subseteq V$ system of vectors

$$\left|\left|\overset{\rightarrow}{\sum_{v \in V"}} v\right|\right| \leq_{\mathbb{R}} \overset{(\mathbb{R})}{\sum_{v \in V"}} ||v||.$$

Obviously

$$\forall (v_1, v_2) \in V \times V : |\ ||v_1|| -_{\mathbb{R}} ||v_2||\ |_{\text{Eucl.}} \leq_{\mathbb{R}} ||v_1 \vec{-} v_2||.$$

We note that the aforementioned four criteria in the definition of norms are redundant, since the nonnegativity of a norm can be proven from the three others by

$$\forall v \in V : 0_{\mathbb{R}} = \|\vec{0}\| = \|v \vec{-} v\| = \|v \vec{+} (\vec{\text{-}}v)\| \leq_{\mathbb{R}} \|v\| +_{\mathbb{R}} \|\vec{\text{-}}v\| = \|v\| +_{\mathbb{R}} \|v\| = 2_{\mathbb{R}} *_{\mathbb{R}} \|v\| \to$$

$$\to \forall v \in V : \|v\| \geq_{\mathbb{R}} 0_{\mathbb{R}}$$

Furthermore half of the the fourth condition is redundant as well, since $\|\vec{0}\| = 0_{\mathbb{R}}$ can be proven from the homogenity condition.

2.3 Finite Fields and Finite Vector spaces

2.3.1 Definition and some easy examples

Since throughout the entire work we will discuss concepts based on finite filelds and finite vector spaces, it is paramount to properly explain them. In this subsection I cite [1], [2] and [6].

Definition Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field. The field F is said to be a **finite field** by definition if and only if

$$\text{Card}(K) \in \mathbb{N}$$

holds. Let $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{\text{-}})$ be a vector space. The vector space W is called a **finite vector space** if and only

$$\text{Card}(V) \in \mathbb{N}$$

is true.

The simplest case of a finite field is

$$\mathbb{F}_2 = (\{0, 1\}, 0, 1, +, *, \text{-}, ^{-1})$$

where

$$0 * 0 = 1 * 0 = 0 * 1 = \text{-}0 = 0 + 0 = 1 + 1 = 0$$

and

$$1 * 1 = 0 + 1 = 1 + 0 = 1^{-1} = \text{-}1 = 1$$

Another simple example for finite fields is

$$\mathbb{F}_3 = (\{0,1,2\}, 0, 1, +, *, \text{-}, ^{-1})$$

where

$$0+0 = 1+2 = 2+1 = \text{-}0 = 0$$
$$1+0 = 0+1 = 2+2 = \text{-}2 = 1$$
$$2+0 = 0+2 = 1+1 = \text{-}1 = 2$$

and

$$0*0 = 0*1 = 1*0 = 0*2 = 2*0 = 0$$
$$1*1 = 2*2 = 1^{-1} = 1$$
$$2*1 = 1*2 = 2^{-1} = 2$$

Further example is a field with four elements

$$\mathbb{F}_4 = (\{0,1,A,B\}, 0, 1, +, *, \text{-}, ^{-1}).$$

Obviously $0+x = x$ for all $x \in \{0,1,A,B\}$ is obvious. Furthermore let $1+1=0$. What will be the value of $A+1$? It certainly cannot be A, because $1=0$ would follow. It cannot be 1 either, because then we would get $A=0$. The value of $A+1$ cannot be 0 either, because then $A=1$ would follow. The only remaining option is $A+1=B$. What would be the value of $A*B$? It cannot be A, because then $B=1$ would follow. For the same reason it cannot be B either. We have already seen that two nonzero element multiplied cannot result in zero, therefore we must define $A*B=1$ if we want to have a field. By following this type of logic we can calculate the following:

$$0+0 = 1+1 = A+A = B+B = \text{-}0 = 0$$
$$1+0 = 0+1 = A+B = B+A = \text{-}1 = 1$$
$$A+0 = 0+A = 1+B = B+1 = \text{-}A = A$$
$$B+0 = 0+B = 1+A = A+1 = \text{-B} = B$$

and

$$0*0 = 0*1 = 1*0 = 0*A = A*0 = 0*B = B*0 = 0$$
$$1*1 = A*B = B*A = 1^{-1} = 1$$
$$A*1 = 1*A = B*B = B^{-1} = A$$
$$B*1 = 1*B = A*A = A^{-1} = B.$$

It can be easily shown that the structure we defined here is indeed a field, and the only field with four elements.

2.3.2 Prime numbers, number of elements in a finite field

From now - for the sake of better readibility- fields will be referred to with a symbol such as $\mathbb{F}$ not making a distinction between the structure and the set containing the elements when it does not cause ambiguity. Furthermore, again for the sake of readibility we will utilize the same notation for addition over $\mathbb{N}$ and a vector space or a field when it does not lead to ambiguity. The same can be said for additive and multiplicative identities as well. In the remaining part of the subsection we will look at some pivotal attributes and concepts of finite fields.

Definition The set

$$\mathbb{P} = \{p \in \mathbb{N} \mid \mathrm{Card}\{k \in \mathbb{N} \mid p \equiv 0 \pmod{k}\} = 2\}$$

is called the set of **prime numbers**.

It is not difficult to show that

$$\mathrm{Card}(\mathbb{P}) = \mathrm{Card}(\mathbb{N}).$$

Theorem 2.13 *Let $\mathbb{F}$ be a finite field. Then*

$$\exists\ (p, k) \in \mathbb{P} \times \mathbb{N}^+ \ :\ \mathit{Card}(\mathbb{F}) = p^k$$

holds.

The finite field with the number of elements q will be denoted as $\mathbb{F}_q$ and the n-dimensional vector space over $\mathbb{F}_q$, furthermore the set containing the vectors will be shortly referred to by the symbol $\mathbb{F}_q^n$.

2.3.3 Multiplicative order, primitive elements

Primitive elements of a finite field will occur in this work, for instance in 5.1.2, so they will be mentioned briefly.

Definition Let $\mathbb{F}_q$ be a finite field. The **multiplicative order** of an element $x \in \mathbb{F}_q$ is defined by

$$\mathrm{Ord}(\mathbb{F}_q, x) = \mathrm{Card}\{x^\gamma \mid \gamma \in \mathbb{N}\}.$$

The element $a \in \mathbb{F}_q$ is called a **primitive element** by definition if and only if

$$\mathrm{Ord}(\mathbb{F}_q, a) = q - 1$$

holds.

For instance

$$\mathrm{Ord}(\mathbb{F}_2, 1) = 1 = 2 - 1$$

which means that $1 \in \mathbb{F}_2$ is a primitive element. Let us see another example over $\mathbb{F}_3$. The powers are $1, 2, 1, 2\ldots$ indicating that

$$\mathrm{Ord}(\mathbb{F}_3, 2) = 2 = 3 - 1$$

consequently $2 \in \mathbb{F}_3$ is a primitive element, but $1 \in \mathbb{F}_3$ is not a primitive element, since

$$\mathrm{Ord}(\mathbb{F}_3, 1) = 1 \neq 2 = 3 - 1.$$

Another example is

$$\mathrm{Ord}(\mathbb{F}_4, A) = 3$$
$$\mathrm{Ord}(\mathbb{F}_4, B) = 3$$

since $A^1 = A$, $A^2 = A * A = B$, $A^3 = A^2 * A = B * A = 1$, $A^4 = A$ and $B^1 = B$, $B^2 = A$, $B^3 = B^2 * B = A * B = 1$, $B^4 = B$. Consequently $A, B \in \mathbb{F}_4$ are primitive elements. Now we will observe some attributes of the finite fields expressed with the notion of multiplicative order and primitive elements.

Theorem 2.14 *Every finite field has a primitive element.*

2.3.4 Construction of $\mathbb{F}_{p^k}$ from polynomials

Theorem 2.15 *For all $(p,k) \in \mathbb{P} \times \mathbb{N}^+$ there is a unique finitie field which contains p^k elements. Let $\mathbb{F}_q$ be a finite field, and $x \in \mathbb{F}_q \setminus \{0\}$. We have*

$$x^{q-1} = 1$$

and

$$q - 1 \equiv 0 \ (mod\ Ord(\mathbb{F}_q, x)).$$

Furthermore we have

$$X^{q-1} - 1 = \prod_{x \in \mathbb{F}_q \setminus \{0\}} (X - x)$$

and

$$X^q - X = \prod_{x \in \mathbb{F}_q} (X - x) = X \prod_{x \in \mathbb{F}_q \setminus \{0\}} (X - x) = X(X^{q-1} - 1)$$

2.3.5 The $\mathbb{F}_p$ field

Despite everyone being familiar with the congruence relation and in spite of the fact that it was already utilized in this work, for the sake of completness we give a formal definition specifically over $\mathbb{Z}$.

Definition Let $\Psi \in \mathbb{Z}$ be an integer. The

$$\equiv (\text{mod } \Psi) \subseteq \mathbb{Z} \times \mathbb{Z}$$

relation defined by the formula

$$\forall (x,y) \in \mathbb{Z} \times \mathbb{Z} : (x,y) \in \equiv (\text{mod } \Psi) \leftrightarrow \Psi \mid x -_{\mathbb{Z}} y$$

is called the modulo Ψ **congruence relation** on $\mathbb{Z}$. If $(x,y) \in \equiv (\text{mod } \Psi)$ then we usually write that

$$x \equiv y \ (\text{mod } \Psi)$$

instead.

One can effortlessly deduce that this relation is an equivalence relation. The equivalence classes are as follow

$$[x]^{\Psi} = \{y \in \mathbb{Z} : y \equiv x \ (\text{mod } \Psi)\}.$$

The partition is

$$\mathbb{Z}/\equiv (\text{mod } \Psi) = \{[x]^{\Psi} : x \in \mathbb{Z}\}.$$

Let now $p \in \mathbb{P}$ be a prime number. Consider the 7-tuple

$$\mathbb{F}_p = (\mathbb{Z}/\equiv (\text{mod } p), [0_{\mathbb{Z}}]^p, [1_{\mathbb{Z}}]^p, +_p, *_p, \text{-}_p, {}^{-1_p})$$

where $[0_{\mathbb{Z}}]^p$ is the equivalence class represented by the element $0_{\mathbb{Z}}$,$[1_{\mathbb{Z}}]^p$ is the equivalence class represented by the element $1_{\mathbb{Z}}$, furthermore

$$+_p : (\mathbb{Z}/\equiv (\text{mod } p)) \times (\mathbb{Z}/\equiv (\text{mod } p)) \to \mathbb{Z}/\equiv (\text{mod } p)$$

$$-_p : (\mathbb{Z}/\equiv (\text{mod } p)) \times (\mathbb{Z}/\equiv (\text{mod } p)) \to \mathbb{Z}/\equiv (\text{mod } p)$$

$$\text{-}_p : \mathbb{Z}/\equiv (\text{mod } p) \to \mathbb{Z}/\equiv (\text{mod } p)$$

$$^{-1_p} : \mathbb{Z}/ \equiv \pmod{p} \setminus \{[0_\mathbb{Z}]^p\} \to \mathbb{Z}/ \equiv \pmod{p}$$

are operations defined by the formulas

$$\forall([x],[y]) \in (\mathbb{Z}/ \equiv \pmod{p}) \times (\mathbb{Z}/ \equiv \pmod{p}) : [x]^p +_p [y]^p = [x +_\mathbb{Z} y]^p$$

$$\forall(x,y) \in (\mathbb{Z}/ \equiv \pmod{p}) \times (\mathbb{Z}/ \equiv \pmod{p}) : [x]^p *_p [y]^p = [x *_\mathbb{Z} y]^p$$

$$\forall[x] \in (\mathbb{Z}/ \equiv \pmod{p}) : -_p[x]^p = [-_\mathbb{Z} x]^p$$

$$\forall[x] \in (\mathbb{Z}/ \equiv \pmod{p}) \setminus \{[0_\mathbb{Z}]^p\} \pmod{p} : ([x]^p)^{-1_p} = [x^{p -_\mathbb{Z} 2}]^p.$$

It can be shown that the aforementioned functions are well-defined and that this 7-tuple forms a field. Furthermore it is not difficult to deduce that for $p \notin \mathbb{P}$ this construction will not form a field.

2.4 Codes, Examples

To comprehend the basics being represented in this subsection, I was solely utilizing [1]. In this subsection we will define codes and see some elementary examples. Additionally, we will meet a special category of codes, the repetition codes.

The symbol $\mathbb{F}_q^n$ shortly signifies the vector space where the field is the finitie field with q elements, the set of vectors is $\mathbb{F}_q^n$ and the operations are defined coordinatewise.

Definition Let us consider the vector space $\mathbb{F}_q^n$, and let $C \subseteq \mathbb{F}_q^n$ be such as

$$\exists k \in \mathbb{N}^+ : \mathrm{Card}(C) = q^k$$

holds. In this case the subset C is said to be by definition a **code** with parameters

$$(n,k) = (n, \log_q(\mathrm{Card}(C))).$$

The finite field $\mathbb{F}_q$ is said to be the **alphabet**. Let

$$\phi : \mathbb{F}_q^k \to \mathbb{F}_q^n$$

be an injection such as

$$\mathrm{Im}(\phi) = \{y \in \mathbb{F}_q^n \mid \exists x \in \mathbb{F}_q^k \ \phi(x) = y\} = C$$

is true. The elements of the set $\mathrm{Im}(\phi) = C$ are called the **codewords**. We say that the elements of $\mathbb{F}_q^n$ are the **words**. The process of applying the function ϕ is termed **encoding**. The parameter n is called the **length** of the code. When it is needed for the sake of preciseness, we can refer to the ordered pair

$$\Phi = (\phi, C)$$

as a code.

One of the most trivial examples of a code can be considered as $q = 2$, $k = 1$, $n = 1$, and our injection $\phi : \mathbb{F}_2^1 \to \mathbb{F}_2^1$ be defined in a way that $\phi(0) = 0$ and $\phi(1) = 1$, meaning that $\phi \equiv id$. If we want to send the message 0 0 1, it will be encoded and sent as 0 0 1. If the message 0 0 0 is received and we assume that there is an error in at least one coordinate because of the transmission, we cannot find out what the original message should have been. Let us see some more elementary examples. let $q = 2$, $k = 1$, $n = 2$. This means that we have an injection $\phi : \mathbb{F}_2^1 \to \mathbb{F}_2^2$. Let us define ϕ in a way that $\phi(0) = (0,0)$ and $\phi(1) = (1,1)$. This means that whenever we see the coordinate 0, we will send $(0,0)$ instead, and whenever it would be 1, the characters $(1,1)$ will be sent. The message 0 0 1 will be encoded and sent in the form 00 00 11. Consqequently if the it is received that 00 00 01, we know that an error has happened, and in the case of exactly one error, either 000 or 001 was the original message. Let us now examine another example. If $q = 2$, $k = 1$, $n = 3$ and $\phi(0) = (0, 0, 0)$ and $\phi(1) = (1, 1, 1)$ the original message 0 0 1 will be sent as 000 000 111. If it is received by us that 000 000 011 and we assume that exactly one error has occured, the only possible orginal message can be 0 0 1. This example clearly shows that codes can have different capabilities in terms of error-detection. This elementary idea will be quantified by the concept of Hamming distance and elaborated on in 2.5. But if the case is that either one or two errors could have occured, we cannot show where the error was, and the original message could easily have been either 0 0 0 or 0 0 1. The pattern becomes somewhat obvious, and we know that we can correct even two errors by $q = 2$, $k = 1$, $n = 2$ and $\phi(0) = (0, 0, 0, 0, 0)$ and $\phi(1) = (1, 1, 1, 1, 1)$. If it is received by us that 00000 00000 00111, then it can be clearly seen that the original message must have been 0 0 1 in the case of at most two errors. But again, if we would like to deal with up to three errors,then the original message either was 0 0 0 or 0 0 1. These codes are called repetition codes and will be the special cases of Reed-Muller codes, which will be introduced in 3. Now the concept of (binary) repetition codes will be formalized.

2.4.1 Repetition codes

Definition Let $q = 2$, $k = 1$,

$$\phi(0) = (0, 0, \ldots 0)$$

and

$$\phi(1) = (1, 1, \ldots 1).$$

In this case our code is called by definition a **repetition code of length** n denoted by $RC(n)$.

Let us compare now two codes with the same length. The first is the repetition code of length three, the second is defined by $q = 2$, $k = 1$, $\phi(0) = (0, 0, 0)$ and $\phi(1) = (0, 0, 1)$. If the received message is in the case of the second code 000 000 000 and it is assumed that at most one error occured, the original message could have been 000, 001, 010, 100. If the received message is 000 000 000 in the case of the repetition code, and it is assumed that at most one error occured, we do know that the number of errors is exactly zero. This elementary example clearly shows the difference of error detecting and correcting capabilities between codes of same length.

2.5 Hamming distance, weight, errors, and some basic consequences

For this subsection I cite [1], [2], [3] [8], [11] and [13].

2.5.1 Metric spaces

Metric spaces are briefly mentioned since they will be appearing a lot in this subsection. The concept of metric spaces and normed vector spaces allow us to later handle codewords as points, meaning that they are paramount in coding theory.

Definition Let (S, d) be an ordered pair, where S is a set and

$$d : S \times S \to \mathbb{R}$$

is a function with two variables. The (S, d) pair is called a **metric space** and d is called a **metric on the set** S by definition if and only if the conditions

$$\forall (x, y) \in S \times S : d(x, y) = 0_{\mathbb{R}} \leftrightarrow x = y$$

$$\forall (x,y) \in S \times S : d(x,y) = d(y,x)$$

$$\forall (x,y,z) \in S \times S \times S : d(x,y) +_{\mathbb{R}} d(y,z) \geq_{\mathbb{R}} d(x,z)$$

are satisfied. (Later we will see that we can write $d : S \times S \to \mathbb{R}^+$ instead.) For the sake of simplicity we can say that the real number $d(x,y)$ is the **distance** of the **points** x and y.

We can see that the third condition is basically the idea of the **triangle inequality**, and roughly/informally speaking reflects on the idea that when we talk about distance, we tend think that the "straight line" is the shortest between two points. Sometimes this condition is referred to as **subadditivity**. The second condition expresses that we want distance to be symmetrical.The presence of the first condition is fairly obvious. From these three conditions it can be proven that the distance of two points can never be negative.

Theorem 2.16 *Let (S,d) be a metric space. We have*

$$\forall (x,y) \in S \times S : d(x,y) \geq_{\mathbb{R}} 0_{\mathbb{R}}.$$

In other words the distance of any two points cannot be negative.

Proof Let us substitute $z = x$ to the triangle inequality. It is obtained that

$$\forall (x,y) \in S \times S : d(x,y) +_{\mathbb{R}} d(y,x) \geq_{\mathbb{R}} d(x,x).$$

Utilizing the condition of symmetry the above condition is transformed into

$$\forall (x,y) \in S \times S : d(x,y) +_{\mathbb{R}} d(x,y) \geq_{\mathbb{R}} d(x,x).$$

From the first condition we know that the right side of the inequality is $0_{\mathbb{R}}$. We get to the inequality

$$\forall (x,y) \in S \times S : 2_{\mathbb{R}} *_{\mathbb{R}} d(x,y) \geq_{\mathbb{R}} 0_{\mathbb{R}}$$

which is equivalent to

$$\forall (x,y) \in S \times S : d(x,y) \geq_{\mathbb{R}} 0_{\mathbb{R}}$$

because $2_{\mathbb{R}}$ is a positive real number. ■

The following theorem will emphasize the connection between norms and distances.

Theorem 2.17 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $W = (V, F, \vec{0}, \vec{+}, \vec{*}, \vec{-})$ be a vector space. Let $(W, \|.\|)$ be a normed space. Let the $d : V \times V \to \mathbb{R}$ function defined as*

$$\forall (v_1, v_2) \in V \times V : d(v_1, v_2) = \|v_2 \vec{-} v_1\|.$$

In this case (V,d) is a metric space. Furthermore

$$\forall v \in V \;\; d(\vec{0}, v) = d(v, \vec{0}) = \|v\|$$

holds as well.

Proof From the definition of d and the definition of norms we have We have

$$d(v_1, v_2) = 0_{\mathbb{R}} \leftrightarrow \|v_2 \vec{-} v_1\| = 0_{\mathbb{R}} \leftrightarrow v_2 \vec{-} v_1 = \vec{0} \leftrightarrow v_1 = v_2.$$

The symmetry condition easily follows from the very fact that the additive inverse of a vector has the same norm as the original vector, so we have

$$d(v_1, v_2) = \|v_2 \vec{-} v_1\| = \|\vec{-}(v_2 \vec{-} v_1)\| = \|\vec{-} v_2 \vec{+} v_1\| = \|v_1 \vec{-} v_2\| = d(v_2, v_1).$$

The triangle inequality comes from the triangle inequality of norms, so

$$d(v_1, v_3) = \|v_3 \vec{-} v_1\| = \|v_3 \vec{-} v_2 \vec{+} v_2 \vec{-} v_1\| \leq_{\mathbb{R}} \|v_3 \vec{-} v_2\| +_{\mathbb{R}} \|v_2 \vec{-} v_1\| = d(v_3, v_2) +_{\mathbb{R}} d(v_2, v_1).$$

This proves that d is really a distance function. For the second claim we have

$$\forall v \in V \;\; d(v, \vec{0}) = d(\vec{0}, v) = \|v \vec{-} \vec{0}\| = \|v\|$$

completing our proof. ■

An example will appear in 2.5.2.

Theorem 2.18 *Let $F = (K, 0, 1, +, *, \text{-}, ^{-1})$ be a field and $|.| : K \to \mathbb{R}$ be an absolute value function over the field. The $d : K \times K \to K$ function defined by*

$$\forall (x, y) \in K \times K : d(x, y) = |y - x|.$$

In this case $(K, |.|)$ is a metric space and we have

$$d(0, x) = d(x, 0) = |x|.$$

Proof Trivial from the previous proof. ■

2.5.2 Hamming distance, minimal distance, Hamming weight, weight of codes

Definition Let us consider the function $d : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{R}$ defined by

$$d(u, v) = \text{Card}\{i \mid u_i \neq v_i\}$$

for every ordered pair. The real number $d(u, v)$ is said to be by definition the **Hamming distance** of u and v. Let $C \subseteq \mathbb{F}_q^n$ be a code. In this case the real number

$$\text{Inf}\{d(u, v) \mid (u \neq v) \wedge (u, v \in C)\}$$

is called by definition the **minimal distance** of the code C and is denoted with the symbol $d(C)$. We will define the function

$$\text{wt} : \mathbb{F}_q^n \to \mathbb{R}$$

by the formula

$$\text{wt}(u) = d(u, 0)$$

which means that $\text{wt}(u)$ is the number of nonzero coordinates in u, and is called the **Hamming weight** of the word. Expressing this by formula we have

$$\text{wt}(u) = \text{Card}\{i \mid u_i \neq 0\}.$$

The number

$$\text{wt}(C) = \text{Inf}\{\text{wt}(u) \mid (\text{wt}(u) \neq 0) \wedge (u \in C)\}$$

is called the **weight of the code** C by definition.

Indeed it is convenient to phrase

$$d(u, v) = \sum_{\forall i} I(u_i \neq v_i) = \sum_{\forall i} (1 - I(u_i = v_i))$$

where I is the indicator function. The wt function for $\mathbb{F}_q^1$ becomes the trivial absolute value function, which is zero for the zero element, and 1 otherwise.

Theorem 2.19 *For all $u \in \mathbb{F}_2^n$ the equality*

$$wt(u + v) = wt(u) + wt(v) - 2wt(u * v)$$

holds, where $*$ is understood as a component-wise multiplication.

The source of this theorem is [12]. The proof was not detailed in the reference literature.

Proof For the sake of clarity and better readibility let the lower index 2^n refer to the operations of $\mathbb{F}_2^n$, the lower index 2 refer to the modulo 2 operations and every operation without a lower index will denote an operation over $\mathbb{R}$. Moreover 0_2 will be the addtive neutral element of $\mathbb{F}_2$ and similarly 1_2 will be the multiplicative neutral element of $\mathbb{F}_2$.

So what is needed to be shown is

$$\forall (u, v) \in \mathbb{F}_2^n \times \mathbb{F}_2^n : wt(u +_{2^n} v) = wt(u) + wt(v) - 2wt(u *_{2^n} v)$$

Let $wt_2 : \mathbb{F}_2 \to \mathbb{R}$ be the weight function restricted to the coordinates, meaning that

$$wt(0_2) = 0$$

and

$$wt(1_2) = 1.$$

The idea is that we need to observe the coordinates separately. First with a chart we will show that

$$\forall (a, b) \in \mathbb{F}_2 \times \mathbb{F}_2 : wt_2(a +_2 b) = wt_2(a) + wt_2(b) - 2wt_2(a *_2 b)$$

a	b	A	B	C	D	E	F	G	H
1	**1**	**0**	1	1	1	0	1	2	0
1	**0**	**1**	0	1	0	1	0	1	1
0	**1**	**1**	0	0	1	1	0	1	1
0	**0**	**0**	0	0	0	0	0	0	0

In the chart the notations

$$A = a +_2 b$$

$$B = a *_2 b$$

$$C = wt_2(a)$$

$$D = wt_2(b)$$

$$E = wt_2(a +_2 b)$$

$$F = wt_2(a *_2 b)$$

$$G = wt_2(a) + wt_2(b)$$

$$H = wt_2(a) + wt_2(b) - 2wt_2(a *_2 b)$$

are utilized. Furthermore for better aesthetics we have used the notations $\mathbf{1} = 1_2$ and $\mathbf{0} = 0_2$. It can be observed easily that

$$\forall u \in \mathbb{F}_2^n : wt(u) = \sum_{i=1}^{n} wt_2(u_i)$$

holds. From this because of the linearity of the mapping $u \mapsto u_i$ and the linerarity of the $\sum$ operator we can see that

$$
\begin{aligned}
wt(u+v) = \sum_{i=0}^{n} wt_2((u +_2 v)_i) = \sum_{i=0}^{n} wt_2(u_i +_2 v_i) = \\
= \sum_{i=0}^{n} (wt_2(u_i) + wt_2(v_i) - 2wt_2(u_i v_i)) = \\
= \sum_{i=0}^{n} wt_2(u_i) + \sum_{i=0}^{n} wt_2(v_i) - \sum_{i=0}^{n} 2wt_2(u_i v_i) = \\
= \sum_{i=0}^{n} wt_2(u_i) + \sum_{i=0}^{n} wt_2(v_i) - 2\sum_{i=0}^{n} wt_2(u_i v_i) = \\
= \sum_{i=0}^{n} wt_2(u_i) + \sum_{i=0}^{n} wt_2(v_i) - 2\sum_{i=0}^{n} wt_2((uv)_i) = \\
= wt(u) + wt(v) - 2wt(u * v)
\end{aligned}
$$

meaning that our proof is complete. ■

2.5.3 Connection with metric spaces and normed vector spaces

It comes without difficulty to observe that the ordered pair $(\mathbb{F}_q^n, d)$ forms a metric space. We can see that

$$d(u,v) = 0 \leftrightarrow Card\{i \mid u_i \neq v_i\} = 0 \leftrightarrow \nexists i : u_i \neq v_i \leftrightarrow \forall i : u_i = v_i \leftrightarrow u = v$$

and because of the simmetry of the $\neq$ relation that

$$d(u,v) = Card\{i \mid u_i \neq v_i\} = Card\{i \mid v_i \neq u_i\} = d(v,u)$$

furthermore

$$d(u,v) = Card\{i \mid u_i \neq v_i\} \leq Card\{i \mid u_i \neq w_i\} + Card\{i \mid w_i \neq v_i\} =$$
$$= d(u,w) + d(w,v)$$

holds. To see why

$$Card\{i \mid u_i \neq v_i\} \leq Card\{i \mid u_i \neq w_i\} + Card\{i \mid w_i \neq v_i\}$$

is true, we will first consider the case of $\mathbb{F}_q^1$. Now we have only one index and one coordinate.The number $Card\{i \mid u_i \neq v_i\}$ could be greater than

$$Card\{i \mid u_i \neq w_i\} + Card\{i \mid w_i \neq v_i\}$$

only if

$$Card\{i \mid u_i \neq v_i\} = 1 \wedge Card\{i \mid u_i \neq w_i\} + Card\{i \mid w_i \neq v_i\} = 0$$

which would imply that

$$Card\{i \mid u_i \neq w_i\} = 0 \wedge Card\{i \mid w_i \neq v_i\} = 0$$

which means that

$$\forall i : u_i = w_i \wedge w_i = v_i$$

from there because we have only one coordinate

$$u_1 = w_1 \wedge w_1 = v_1$$

from where the transitivity of equality it is obtained that

$$u_1 = v_1$$

implying in our case that

$$\forall i : u_i = v_i$$

which result in

$$Card\{i \mid u_i \neq v_i\} = 0$$

contradicting the initial assumption that

$$Card\{i \mid u_i \neq v_i\} = 1.$$

Now we can easily utilize similar thoughts for $\mathbb{F}_q^n$ for $n > 1$, we just have to consider the vectors by individual coordinates, thereby reducing the problem to that of $\mathbb{F}_q^1$. The

$$\forall (u, v) : d(u, v) \geq 0$$

fact comes from the triangle inequality and the symmetry proterty, but can be seen by the nonnegativity of cardinality as well. Furthermore can be concluded that

$$d(u, v) = wt(u - v)$$

is an universal equality. Because of the linearity of the function $u \mapsto u_i$ we can deduce that

$$d(u, v) = Card\{i \mid u_i \neq v_i\} = Card\{i \mid u_i - v_i \neq 0\} =$$

$$= Card\{i \mid (u - v)_i \neq 0\} = d(u - v, 0) = wt(u - v)$$

holds. Additionally, we can clearly see that we could have defined d by simply utilizing the aforementioned formula and defining wt beforehand.This can remind us to the fact that if $(V, \|.\|)$ is a normed space, then the

$$\forall (x, y) : d(x, y) = \|x - y\|$$

formula defines a distance function. Now let us examine why exactly is wt a norm over $\mathbb{F}_q^n$. Because of the fact that $(\mathbb{F}_q^n, d)$ forms a metric space we obtain

$$wt(u) = 0 \leftrightarrow d(u, 0) = 0 \leftrightarrow d = 0.$$

Now utilizing the fact that translation both points with a vector does not alter their distance and we get that

$$wt(u + v) = d(u + v, 0) = d(u + v - v, -v) = d(u, -v) \leq$$

$$\leq d(u, 0) + d(0, -v) = d(u, 0) + d(-v, 0) = wt(u) + wt(-v) = wt(u) + wt(v).$$

Furthermore for any nonzero scalar $\lambda \in \mathbb{F}_q$

$$wt(\lambda u) = d(\lambda u, 0) = d(u, 0) = wt(u) = |\lambda|_{triv} u$$

where

$$|.|_{triv} : \mathbb{F}_q \to \mathbb{R}_0^+$$

is the trivial absolute value function. Indeed, any norm is homogeneous with respect to exactly one absolute value function if the vector space contains at least two vectors.. The fact that wt

is only homogeneous with respect to $|.|_{triv}$ could be easily seen from the fact that restricted to $\mathbb{F}_q^1$ the function wt gives us the trivial absolute value function. The nonnegativity comes from the triangle inequality and the facts that $wt(0) = 0$ and $wt(u) = wt(-u)$, but can be easily seen from the nonnegativity of cardinality and from the fact that wt has been defined by using the Hamming distance. As an example, look at

$$d(RC(n)) = wt(RC(n)) = n$$

since

$$d((0, 0, \ldots 0), (1, 1, \ldots, 1)) = n$$

obviously holds. As another example

$$d(\mathbb{F}_q^n) = (\mathbb{F}_q^n) = 1$$

since for all unit vector $e \in \mathbb{F}_q^n$ it holds that

$$wt(e) = 1$$

and the weight of a code cannot be zero. From the definition of the weight of a code it can be seen that $wt(C) \neq 0$. This also means that

$$wt(\{0\}) = Inf\{wt(u) \mid (wt(u) \neq 0) \wedge (u \in \{0\})\} =$$

$$= Inf\{wt(u) \mid (wt(u) \neq 0) \wedge (u = 0)\} =$$

$$= Inf\{wt(u) \mid (wt(u) \neq 0) \wedge (wt(u) = 0)\} = Inf(\emptyset)$$

does not exist. It comes naturally that we can infer everything to wt which generally holds for all norms, for example

$$wt(u + v) \geq |wt(b) - wt(a)|_{Eucl.}$$

where $|.|_{Eucl.}$ is the $\mathbb{R} \to \mathbb{R}_0^+$ Euclidean absolute value function.

Theorem 2.20 *Let $\mathbb{F}_q^n$ be vector space over a finite field and $(u^{(i)})_{i=1}^h$ a system of words. The inequality*

$$wt\left(\sum_{i=1}^{h} u^{(i)}\right) \geq \sup_{(0 \leq j \leq h) \wedge (e_{i,j}=0 \leftrightarrow i=j) \wedge (e_{i,j}=1 \leftrightarrow i \neq j)} \sum_{i=1}^{h} (-1)^{e_{j,i}} wt(u^{(i)})$$

holds.

We note that instead of wt we could have written any norm over any normed vector space meaning that

$$\left\|\sum_{i=1}^{h} u^{(i)}\right\| \geq \sup_{(0 \leq j \leq h) \wedge (e_{i,j}=0 \leftrightarrow i=j) \wedge (e_{i,j}=1 \leftrightarrow i \neq j)} \sum_{i=1}^{h} (-1)^{e_{j,i}} \left\|u^{(i)}\right\|.$$

These simple we conclude by generalizing the formula well-known by everyone from high school

$$\forall (x, y) \in \mathbb{R} \times \mathbb{R} : |x - y| \geq ||x| - |y|| = \max(|x| - |y|, |y| - |x|).$$

Proof What we have to notice is that the triangle inequality can be generalized like

$$wt\left(\sum_{i \in I} v^{(i)}\right) \leq \sum_{i \in I} wt(v^{(i)})$$

where $I \subseteq$ is a finite set of indices and $(v^{(i)})_{i \in I}$ is a system of words.

$$wt(u^{(j)}) = wt\left(u^{(j)} + \sum_{1 \le i \le h \wedge i \ne j} u^{(i)} - \sum_{1 \le i \le h \wedge i \ne j} u^{(i)}\right) =$$

$$= wt\left(\sum_{1 \le i \le h} u^{(i)} - \sum_{1 \le i \le h \wedge i \ne j} u^{(i)}\right) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right) + wt\left(-\sum_{1 \le i \le h \wedge i \ne j} u^{(i)}\right) =$$

$$= wt\left(\sum_{1 \le i \le h} u^{(i)}\right) + wt\left(\sum_{1 \le i \le h \wedge i \ne j} u^{(i)}\right) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right) + \sum_{1 \le i \le h \wedge i \ne j} wt(u^{(i)})$$

from where because of the transitivity of inequality and equality it is obtained that

$$wt(u^{(j)}) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right) + \sum_{1 \le i \le h \wedge i \ne j} wt(u^{(i)})$$

which is equivalent with the inequality

$$wt(u^{(j)}) - \sum_{1 \le i \le h \wedge i \ne j} wt(u^{(i)}) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right)$$

and can be expressed as

$$\sum_{1 \le i \le h} (-1)^{e_{i,j}} wt(u^{(i)}) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right)$$

where

$$e_{i,j} = 0 \leftrightarrow i = j \wedge e_{i,j} = 1 \leftrightarrow i \ne j$$

holds. Since we did not use any special trait of j

$$\forall (1 \le j \le h) : \sum_{1 \le i \le h} (-1)^{e_{i,j}} wt(u^{(i)}) \le wt\left(\sum_{1 \le i \le h} u^{(i)}\right)$$

is true. But if

$$wt\left(\sum_{1 \le i \le h} u^{(i)}\right)$$

is greater than or equal to all those expressions, it is greater than or equal to the maximum of those as well meaning that

$$wt\left(\sum_{i=1}^{h} u^{(i)}\right) \ge \sup_{(0 \le j \le h) \wedge (e_{i,j} = 0 \leftrightarrow i = j) \wedge (e_{i,j} = 1 \leftrightarrow i \ne j)} \sum_{i=1}^{h} (-1)^{e_{j,i}} wt(u^{(i)})$$

holds, which we wanted to show.

2.5.4 Spheres, error-detecting and error-correcting

Definition For a triplet

$$(q, n, \alpha) \in \mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}$$

and for a given $u \in \mathbb{F}_q^n$ we can define the **sphere**

$$\text{Sphere}(q, n, \alpha, u) = \{v \in \mathbb{F}_q^n \mid d(u, v) \le \alpha\}.$$

This concept will appear in the future when proving the Hamming bound in 2.5.5. The fact that we introduced these basic concepts so far allows us to formalize and elaborate on the error-detecting and error-correcting capabilities of a code.

Definition Let C be a code and θ, θ' be a positive integer. The code C is ad definitionem called to be **θ-error-detecting** if and only if for every word in C if altered in at least 1 but at most θ coordinates, the resulting word is not a codeword. The code C is by definition said to be **exactly-θ-error-detecting**, if and only if C is θ-error-detecting and C is not $(\theta+1)$-error-detecting. The code C is called **θ'-error-correcting** if and only if for every (u, v) pair of distinct words each altered in θ' number of positions the two resulting words cannot be the same element of $\mathbb{F}_q^n$. The code C is by definition **exactly-θ-error-correcting** if and only if C is θ-error-correcting and not $(\theta+1)$-error correcting.

In the following theorem we will summarize some elementary results about error-detecting and error-correcting capabilities. Furthermore, we will see exactly how the the connection between the concept of Hamming distance and that of the error-detecting and correcting are formulated.

Theorem 2.21 *Let C be a code and θ, θ' be positive integers.*

1. *The code C is θ-error-detecting if and only if*

$$d(C) \geq \theta + 1$$

and θ'-error-correcting if and only if

$$d(C) \geq 2 * \theta + 1.$$

2. *The code C is exactly $(d(C)-1)$-error-detecting and exactly*

$$\lfloor (d(C)-1)/2 \rfloor$$

-error-correcting.

Proof 1. Let code C be such as $d(C) \geq \theta+1$. The smallest possible distance between two words in C is $\theta+1$, consequently altering any word in at least 1 and at most θ positions cannot result in a word from C, leaving us with the conclusion that C is θ-error-detecting. Now let us assume that C is θ-error-detecting. If C is θ-error-detecting, then altering any words in at least 1 and at most θ coordinates cannot generate a new word in C, therefore the minimal distance of C is at least $\theta+1$. Now we will prove the second statement. Let us first assume that $d(C) \geq 2*\theta+1$. If u and u' are such as altering both in at least1 and at most θ coordinates result in the code word u'', we have that

$$d(u, u'') + d(u', u'') \leq \theta + \theta = 2 * \theta.$$

By the utilization of the triangle inequality and the transitivity of inequality we consequently deduce that

$$d(u, u') \leq 2 * \theta,$$

contradicting the fact that the minimal distance in C is at least $2*\theta+1$. This contradiction proves that C is θ-error-correcting. Now we assume that C is θ-error-correcting. If the minimal distance of C were at most $2*\theta$, then by definition there were an ordered pair (u, u') such as $d(u, u') = d \leq 2*\theta$. One can deduce without difficulties that we can alter θ coordinates of u and $d-\theta \leq \theta$ coordinates of u' in such a way, that we result in the same word. This contradiction shows us that the minimal distance of C is at least $2*\theta+1$.

2. These statements immediately follow from 1., if we substitute "$\leq$" to "$=$" and we reach the desired equalities. ■

It becomes clear that the repetition code of length is ecactly $(n-1)$-error-detecting and $\lfloor (n-1)/2 \rfloor$-error-correcting. The code $\{000, 001\}$ is exactly 0-error-detecting and exactly 0-error-correcting.

2.5.5 Hamming bound and Singleton bound

The following theorem will quantify the connection between certain code parameters. The theorem will consist of two really elementary inequalities, featuring $d(C)$, q, $Card(C)$, n. After the theorem we will examine why establishing those inequalities are important, furthermore we will derive some other concepts based on their results.

Theorem 2.22 *Let $C \subseteq \mathbb{F}_q^n$ be a code. For every $\alpha \leq \lfloor (d-1)/2 \rfloor$ we have*

$$q^{n-k} = \frac{q^n}{q^k} = \frac{Card(\mathbb{F}_q^n)}{Card(C)} \geq \sum_{i=0}^{\alpha} \binom{n}{i} (q-1)^i.$$

Furthermore the inequality

$$q^{n-k} = \frac{q^n}{q^k} = \frac{Card(\mathbb{F}_q^n)}{Card(C)} \geq q^{d(C)-1}$$

holds as well.

The first inequality is called the **Hamming bound** and the second one is the **Singleton bound**.

Proof Let us consider the spheres

$$\text{Sphere}(q, n, \alpha, u) = \{v \in \mathbb{F}_q^n \mid d(u, v) \leq \alpha\}.$$

Because of the fundamental properties of the floor function and those of the inequality relation we obtain that $\alpha \leq \lfloor (d-1)/2 \rfloor$ is equivalent with the inequality $2 * \alpha < d$. It means that the radii of the spheres are actually so small that even the two closest centered ones cannot reach each other. By the aforementioned observation, we can easily obtain that the spheres are actually pairwise disjoint. Of course we have exactly $Card(C)$ of those spheres. It is obvious that in $\mathbb{F}_q^n$ the points contained by a sphere is not depending on the center of the sphere, only on the radius. Let us introduce the notion

$$w(\alpha) = \text{Card}(\text{Sphere}(q, n, \alpha, u)).$$

Those spheres do not necessarily cover all the points in $\mathbb{F}_q^n$, consequently

$$\text{Card}(C) * \text{Card}(\text{Sphere}(q, n, \alpha, u)) \leq \text{Card}(\mathbb{F}_q^n).$$

Rearranging this inequality the form

$$\frac{\text{Card}(\mathbb{F}_q^n)}{\text{Card}(C)} \geq \text{Card}(\text{Sphere}(q, n, \alpha, u))$$

is obtained. We will now show that the $\text{Sphere}(q, n, \alpha, u)$ contains exactly

$$\sum_{i=0}^{\alpha} \binom{n}{i} (q-1)^i$$

points, and then the proof of the first inequality is complete. From the center of the sphere u, we can get the other points of the sphere by simply altering i coordinates of, where $i \leq \alpha$. For every coordinate we have q possible values, therefore there is $q-1$ possibilities for the coordinates to be transformed to, which is altogether $(q-1)^i$ options. But we also has to choose which coordinates are to be altered, leaving us with $\binom{n}{i} * (q-1)^i$ options. By the summation of these expressions the desired form is obtained, completing the proof of the first inequality. Now we are to verify the second inequality. We will prove that

$$\mathrm{Card}(C) \leq q^{n-d(C)+1},$$

and from this form by simpy rearranging the inequality we obtain the original statement. So we need to show that the code C contains at most $q^{n-d(C)+1}$ codewords. Let

$$C' \subseteq \mathbb{F}_q^{n-d(C)+1}$$

be a code for which there is a $f : C \to C'$ function such as $f(c) \in C'$ is composed of the last $n-d+1$ coordinates of c. We will show that f is a bijection, consequently

$$\mathrm{Card}(C) = \mathrm{Card}(C') = q^{n-d(C)+1},$$

thereby completing the proof. To reiterate what has been said, the only thing remaining to be proven is that f is a bijection. It can be easily inferred that f is surjective, since the $c' \in C'$ codeword is in the image of f, because if c is such as its first $d-1$ coordinates are 0, and the other coordinates are exactly the same in c and c', we have $f(c) = c'$. But it can be concluded that f is an injection as well, since if there were different u and v codewords in C such as $f(u) = f(v)$, then $f(u)$ and $f(v)$ would be the same in $n-d+1$ positions, consequently u and v could only differ in

$$n - (n - d(C) + 1) = d(C) - 1$$

positions, thereby implying $d(u, v) = d-1$, which would in turn contradict to the fact that $d(C)$ is by definition the smallest possible distance in C. ■

2.5.6 Understanding Hamming and Singleton bounds, the concept of perfect codes

The importance of the aforementioned bounds are connected with the following considerations:

1. We want a code to possess good error-correcting and error-detecting capabilities. Consequently, we need that the distance of the code be greater. We have already seen in the proof that
$$\mathrm{Card}(C) = q^k \leq q^{n-d(C)+1},$$
which is equivalent with the inequality
$$k \leq n - d(C) + 1,$$
and when rearranged we are lead to
$$d(C) \leq n - k + 1.$$
One can conclude without difficulties that for a code with great distance n must be great relative to k. It is easy to observe that the previous inequality follows immediately from the Singleton bound.

2. We want the platform to be utlizied with high efficacy. Since the length of the message in the general case becomes longer by n/k, it can be inferred that we are seeking for a code when n is relatively small compared to k.

When considering the aforementioned two desires, we can see that those ideas clearly appear in the bounds proven above. Now we will briefly examine the case when in the bounds equality holds. When equality holds in the Hamming or in the Singleton bound, we see that $n-k$ is maximalized, which in turn implies that the code under consideration follows the first desire, roughly speaking we have a good code in terms of error-detecting and error-correcting capabilities. Furthermore, we have that the union of the spheres are containing all points of $\mathbb{F}_q^n$.

Definition Let $C \subseteq \mathbb{F}_q^n$ be a code. We say that C is a **perfect code** by definition if and only if the union of the spheres mentioned in the proof of the Hamming bound are containing every point of $\mathbb{F}_q^n$. The **rate** of a code is $R = \frac{k}{n}$.

We can easily deduce that a θ-error-correcting code is perfect code if and only if for every word there exists a codeword not further than θ.

2.6 Linear codes, generator matrices, controll matrices, dual codes

In this subsection, my sources were [1], [10] and [12].

2.6.1 Linear codes, generator matrices, controll matrices

Definition Let $C \subseteq \mathbb{F}_q^n$ be a code. We call C a **linear code** by definition if and only if C is a subspace of $\mathbb{F}_q^n$. The matrix $G_C \in \mathbb{F}_q^{n*k}$ is called the **generator matrix** of the linear code C by definition if and only if

$$C = \{G_C * u \mid u \in \mathbb{F}_q^n\},$$

in other words if

$$C = \mathrm{Im}(G_C).$$

The matrix $P_C \in \mathbb{F}_q^{(n-k)*n}$ is said to be the by definition **controll matrix** of the linear code C if and only if

$$C = \{u \in \mathbb{F}_q^n \mid P_C(u) = 0\},$$

or equivalently if

$$C = \mathrm{Ker}(P_C).$$

The **dimension** of the linear code C is by definition the dimension of the subfield $C \leq \mathbb{F}_q^n$. Linear codes are often referred to by the triplet featured with lower index

$$C = [n, k, d(C)]_q = [n, \log_q Card(C), d(C)]_q.$$

Note that in this case C is a vector space, and therefore not only the (C, d) pair is a metric space but (C, wt) is a normed vector space as well. We can immediately observe from the definition that a matrix with the proper dimensions A is a generator matrix of code C if and only if the columns of A form a basis of C.

2.6.2 Weight and distance of linear codes

Now we will highlight one of the most important attribute of linear codes, then we will explore some elementary connections between generator matrices and controll matrices.

Theorem 2.23 *For a linear code C we have $d(C) = wt(C)$.*

I understood this basic concept from [12] and relied solely on it as a source.

Proof By the definition of the weight of C we know that there are words u and v in C that $d(u, v) = d(C)$. Since C is a linear code, $s = u - v$ is also a codeword in C. Because of the equality chain

$$wt(s) = wt(u - v) = d(u, v) = d(C)$$

we know that there is a codeword s in C with a weight at of $d(C)$, consequently we obtain that the weight of the whole code is at most $d(C)$, in other words $wt(C) \leq d(C)$. Now we only have to prove that $wt(C) \geq d(C)$. By the definition of the weight of a code there is a codeword u in C such as

$$wt(C) = wt(u) = d(u, 0).$$

Since C is a linear code, 0 is contained by C. According to the equality above we have two codewords in C with a distance $wt(C)$, meaning that $wt(C) \geq d(C)$. ■

2.6.3 Dual and self-dual codes

Now we will define and elaborate on a concept which is highly attached to that of generator and controll matrices.

Definition Let $C \subseteq \mathbb{F}_q^n$ be a linear code. The linear code $C^\perp$ is called by definition the **dual code** of the code C. The linear code C is said to be **self-dual** by definition if and only if C is a dual code of C.

Theorem 2.24 *Let $C \leq \mathbb{F}_2^n$ be a a linear code. If C is self-dual, then*

$$n \equiv 0 \ (mod\ 2)$$

holds. Furthermore

$$Card\{u \in C \mid wt(u) \equiv 0 \ (mod\ 4)\} = \left(\frac{3}{4} \pm \frac{1}{4}\right)n$$

is true as well.

Proof Because of the definition of self-dual codes we have $C = C^\perp$ from where

$$Dim(C) = Dim(C^\perp)$$

obviously follows. It is also trivial that

$$Dim(C) + Dim(C^\perp) = n.$$

Combining these two aforementioned facts it is obtained that

$$n = Dim(C) + Dim(C^\perp) = Dim(C) + Dim(C) = 2Dim(C)$$

and by the transitivity of equality

$$n = 2Dim(C)$$

implying that

$$n \equiv 0 \pmod{2}.$$

Now we will prove the second claim. If

$$C' = \{u \in C \mid wt(u) \equiv 0 \pmod{4}\} = C$$

then the proof is complete. If

$$\exists u \in C : wt(u) \not\equiv 0 \pmod 4$$

then consider that all codewords either belong to C' or $u + C'$. Now it is clear that all codewords in C have even weight. The ones which have a weight divisible by 4 will go to C' and the others to $u + C'$ meaning that we have found a partition of C. It implies that C' is a subgroup of C with index 2, containing exactly half of the words. ■

It can be easily seen that the statement does not hold for any $\mathbb{F}_q^n$. For example in $\mathbb{F}_5^2$ the code generated by $(1, 2)$ is self dual since

$$\langle \lambda_1(1,2), \lambda_2(1,2) \rangle = \lambda_1\lambda_2\langle (1,2),(1,2) \rangle = \lambda_1 * \lambda_2 * 0 = 0$$

and because of

$$C = \{(0,0),(1,2),(2,4),(3,1),(4,3)\}$$

it can be seen that all the nonzero codewords have a weight of 2. This fact could be easily observed from

$$\forall \lambda \neq 0 : wt(\lambda(1,2)) = wt(1,2) = 2$$

as well.

Theorem 2.25 *Let $C \leq \mathbb{F}_q^n$ be a linear code, and let $wt(C) = 1$. In this case the statement*

$$\nexists v \in c^{\perp} : wt(v) = n$$

holds.

Proof Since $wt(C) = 1$, there exists a codeword $u \in C$ such that

$$wt(u) = wt(C) = 1$$

is true. In this case there exists uniquely a $1 \leq i \leq n$ index and a $0 \neq \lambda \in \mathbb{F}_q$ scalar such that $u = \lambda e_i$, where e_i is the vector whose i coordinate is 1 and the other ones are 0. Let us indirectly assume that there is a $v \in C^{\perp}$ codeword such that $wt(v) = n$, Now there uniquely exists a sequence of $(\lambda_i)_{j=1}^n \in \mathbb{F}_q^n$ scalars such that

$$v = \sum_{j=0}^{n} \lambda_j e_j$$

and

$$\nexists j : \lambda_j = 0$$

hold. Since C and $C^{\perp}$ are by definition orthogonal to each other, the scalar product of u and v should be 0. Now it can be obtained that

$$\langle u, v \rangle = \left\langle \lambda e_i, \sum_{j=0}^{n} \lambda_j e_j \right\rangle = \sum_{j=0}^{n} \left\langle \lambda e_i, \lambda_j e_j \right\rangle =$$

$$= \sum_{j=0}^{n} \lambda\lambda_j \langle e_i, e_j \rangle = \lambda\lambda_i \langle e_i, e_i \rangle = \lambda\lambda_i$$

from which because u and v are orthogonal to each other we get $\lambda\lambda_i = 0$. It means that $\lambda = 0 \vee \lambda_i = 0$. It cannot be the case that $\lambda = 0$, since $wt(u) = 1$ and $\lambda = 0$ is impossible, since $wt(v) = n$. This contradiction means that the indirect assumption is false, completing our proof.

Theorem 2.26 *Let C be a linear code with parameters (n, k).*

1. *The matrix A is a controll matrix of the code C if and only if A^T is a generator matrix of $C^\perp$.*
2. *For every generator matrix and controll matrix of C we have*
$$P_C * G_C = 0.$$
3. *The matrix $A \in \mathbb{F}_q^{(n-k)*n}$ is a controll matrix of C if and only if $Rank(A) = n - k$ and $A * G_C = 0$.*
4. *The matrix $A \in \mathbb{F}_q^{n*k}$ is a generator matrix of C if and only if $Rank(A) = k$ and $P_G * A = 0$.*
5. *The codes C_1 and C_2 are duals of each other if and only if*
$$G_{C_1}^T * G_{C_2} = 0$$
and $Rank(G_{C_1}) + Rank(G_{C_2}) = n$.
6. *The code C is self-dual if and only if $G_C * G_C^T = 0$.*
7. *The code C is self-dual if and only if $P_C * P_C^T = 0$.*

Proof 1. First let us assume that A is a controll matrix of C. In this case every word in C is orthogonal to every row of A, therefore the matrix A only contains rows, which are orthogonal to words in C. But A is a controll matrix of C, which means that it has $n - k$ linearly independent rows. All these rows are from $C^\perp$, and $Dim(C^\perp) = n - k$, which means that the rows of A form a basis of $C^\perp$. This means that the columns of A^T form a basis of $C^\perp$ But we have already seen that having basis as columns means a generator matrix, in this case for the subspace $C^\perp$. All implications are indded equivalences in the aforementioned proof, meaning that the converse is also proven.

2. Let G_C be a generator matrix of C and P_C be a controll matrix of C. Because of the first statement of the theorem we know that P_C^T is a generator matrix of $C^\perp$, implying that the columns of P_C^T form a basis of $C^\perp$, rendering the rows of P_C to be a basis of $C^\perp$. But the columns of the generator matrix G_C form a basis of C by definition, causing the rows of P_C and the columns of G_C to be pairwise orthogonal with each other. Considering that in the result of $P_C * G_C$ there are only the pairwise scalar products of the rows of P_C and the columns of G_C, we are forced to conclude that $P_C * G_C = 0$.

3. If A is a controll matrix of C, then A^T is a generator matrix of $C^\perp$, which means that
$$n - k = Dim(C^\perp) = Rank(A^T) = Rank(A).$$
Additionally, $A * G_C = 0$ follows because of the second point of the theorem. If $Rank(A) = n - k$, we have that A has a maximal rank. Furthermore if $A * G_C = 0$, we obtain that the rows of A are pairwise orthogonal to the columns of G_C, implying that the rows of A form a linearly independent system of vectors in $C^\perp$. But considering the full rank of A we obtain that the rows of A form a basis of $C^\perp$, which means that the columns of A^T are a basis for $C^\perp$, therefore A^T is a generator matrix of the code $C^\perp$. Because of the first point of the theorem we can easily obtain that A is a controll matrix of the code C.

4. Very similar to the proof of 3.

5. Let C_1 and C_2 be duals of each other. It is clear that
$$G_{C_1}^T * G_{C_2} = (P_{C_2}^T)^T * G_{C_2} = P_{C_2} * G_{C_2} = 0.$$
The converse follows easily as well.■

The statements 6. and 7. follow easily from the previous ones.

2.6.4 Connection between distance and linear independence

Now we will explore the connection between the distance of the code and the linearly dependent and independent columns of the controll matrix.

Theorem 2.27 *Let C be a linear code, and P_C the controll matrix of C.*

1. *The distance of C is at least d if and only if every system of $d-1$ columns of P_C forms a linearly independent system.*
2. *The distance of C is exactly d if and only if the smallest system of columns to be linearly dependent in P_C contains d columns.*
3. *The distance of C is at most d if and only if there exists a system of d columns in P_C which form a linearly dependent system.*

Proof We only have to consider, that if we have a word with a weight x, then x rows of P_C will be linearly dependent. It comes immediately after taking into account that by definition $P_C * v = 0$, and while multiplying we create linear combinations of the columns. Moreover it can be inferred that there is a codeword with weight x if and only if there is a system of x linearly dependent columns in P_C.

1. Let us assume that the distance of C is d. If there were a system of $d-1$ columns which were linearly dependent, then there would be a codeword with weight $d-1$, which is impossible, since the minimum distance of C is d by the assumption.

2. If the distance of the code is exactly d, then of course there is a codeword with distance d, meaning that there will be a system of d linearly dependent columns in P_C. And it will be the smallest one, because if there were a system of dependent columns containing fewer than d vectors, then there were a code with a distance smaller than d, contradicting the initial assumption. Conversely, if the smallest system of columns to be linearly dependent in P_C contains d vectors, then there is a codeword with weight d, but there cannot be codeword with smaller weight, implying that the distance of C is d.

3. Let us assume that the distance of C is at most d. If all system of d columns in P_C formed a linearly independent system, then the smallest possible dependent system would contain at least $d+1$ columns, meaning that the smallest weight among the codewords is at least $d+1$. But it would contradict to the initial assumption that the distance of the code is d. Conversely if there exists a system of d columns which is linearly dependent, then there is a codeword with weight d, meaning that the minimal distance of the code cannot be greater than d. ■

2.6.5 Parameters of a specific family of codes

Theorem 2.28 *Let $C_1 = [n, k_1, d(C_1)]_q$ and $C_2 = [n, k_2, d(C_2)]_q$ be two codes. Let us define*

$$C = \{(u, u+v) \mid (u \in C_1) \wedge (v \in C_2)\}.$$

In this case we have a code

$$C = [2n, k_1 + k_2, min(2d(C_1), d(C_2))]_q.$$

Proof 1. Since we concatenate two codewords each containing n characters, we have that the new word does contain $n + n = 2n$ characters.

2. The codeword $c \in C$ has a form $c = (u, u + v)$. We have q^{k_1} options for u and q^{k_2} options for v, consequently the number of overall choices are

$$q^k = Card(C) = Card(C_1 \times C_2) = q^{k_1} q^{k_1} = q^{k_1+k_2},$$

implying that $k = k_1 + k_2$.

3. First we will show that $d \geq min(2d(C_1), d(C_2))$. For this to happen it is sufficient to verify that all nonzero weight in C in greater or equal than $min(2d(C_1), d(C_2))$. Let $(u, u + v)$ be a nonzero codeword in C. If $v = 0$ then u must be nonzero. In this case

$$wt((u, u + v)) = wt((u, u)) = 2wt(u) \geq 2wt(C_1) =$$

$$= 2d(C_1) \geq min(2d(C_1), d(C_2)).$$

If v is nonzero

$$wt((u, v)) = wt(u) + wt(u + v) \geq wt(u) + (wt(v) - wt(u)) =$$

$$= wt(v) \geq min(2d(C_1), d(C_2)).$$

Now we only need to verify that $d \geq min(2d(C_1), d(C_2))$, and we will arrive to our desired conclusion. By definition there exists an $u \in C_1$ such as $wt(u) = d(C_1)$, and a $v \in C_2$ such as $wt(v) = d(C_2)$. The words (u, u) and $(0, v)$ are codewords in C, and $wt((u, u)) = 2d(C_1)$ and $wt((0, v)) = d(C_2)$, implying that the smallest possible weight is at most

$$min(2d(C_1), d(C_2)),$$

which completes our proof. ■

2.7 Standard form, code parameters, equivalence of codes

I would like to cite [1] as my sole source for this subsection.

Definition The codes $C, C' \subseteq \mathbb{F}_q^n$ are said to be **equivalent** codes by definition if and only if C' can be derived from C by permutating the coordinates and then multiplying specific coordinates by a non-zero scalar. Let G_C be a generator matrix of the code C. The matrix G_C is by definition in the **standard form** if there is a matrix X such as

$$G_C = \begin{pmatrix} E_k \\ X \end{pmatrix}$$

holds. The P_C controll matrix is said to be in the **standard form** by definition if there exists a matrix Y such as

$$P_C = \begin{pmatrix} Y & E_{n-k} \end{pmatrix}$$

We can easily observe from the definition that by permutating the coordinates and multiplying with a non-zero scalar does not alter n, k, and $d(C)$. In the following we will represent how the newly introduced concepts are related to each other.

Theorem 2.29 *For every code C uniquely exists a code C' such as there is a generator matrix $G_{C'}$ which is in standard form and C is equivalent with C'.*

Proof Utilizing Gaussian elimination we obtain the reduced column echelon form of G_C, then we permutate the columns in order to obtain the form mentioned in the definition of standard form. ■

Now we will see a method to create a controll matrix from a generator matrix.

Theorem 2.30 *Le C be a code.If the*

$$G_C = \begin{pmatrix} Id_k \\ X \end{pmatrix}$$

generator matrix is in standard form, then the matrix

$$A = \begin{pmatrix} -X & E_{n-k} \end{pmatrix}$$

is a controll matrix of C in standard form.

Proof One can easily see that $A * G_C = 0$. We also can observe that $Rank(A)$ is maximal, meaning that A is controll matrix of the code C. The matrix is obviously in standard form.■

2.8 Polinomials and Codes

As a source I refer to [1] for this subsection.

There is a conspicuous connection between linear codes and certain polynomials. Let $[n, k, d(C)]_q \leq \mathbb{F}_q^n$ be a linear code.The mapping

$$v = (v_1 \dots v_n) \mapsto \sum_{i=0}^{n-1} v_{i+1} x^{i+1}$$

is between two vector spaces which are isomorphic with each other, consequently we can regard the codewords and the corresponding polynomials as conceptually the same.

Definition Let $g(x) \in \mathbb{F}_q[x]$ be a fixed polynomial and

$$deg(g(x)) = m - n.$$

Let $U \leq \mathbb{F}_q[x]$ denote the subspace of polynomials with degree at most n, including the zero polynomial. It is easy to see that the function

$$W : U \to \mathbb{F}_q[x]$$

defined by

$$W : u \mapsto ug$$

is an injective linear mapping, consequently $Im(W)$ will define a linear subspace of polinomials of $\mathbb{F}_q[x]$ with a degree of at most m, with

$$Dim(Im(W)) = n.$$

Thus we obtained a **polynomial code** $Im(W)$ with a **generator polynomial** W.

Theorem 2.31 *Let $C = Im(W)$ the polynomial code generated by the polynomial g. Let $\mathbb{F}_q \leq F$ be a field extension, and $\alpha \in F$ is such that $Ord_* \alpha \geq m$. If g is such that there exists a number $d \leq m$ and a number j that the consecutive powers*

$$\alpha^j \dots \alpha^{j+d-2}$$

are roots of g, we have $d(C) \geq d$.

Proof Since now the codewords are the polynomials themselves, the Hamming distance is obviously calculated by the coefficients. We must verify that for every nonzero codeword the ug polynomials have at least d nonzero coefficients. The elements $\alpha^j \dots \alpha^{j+d-2}$ are roots of g, therefore they will be roots of the ug polynomials as well. We will indirectly assume that there exitsts a nonzero polynomial code with nonzero coefficients fewer than d, and we will get to a contradiction, proving our initial claim. Let v be a polynomial code

$$v = \sum_{i=0}^{l} v_i^{m_i},$$

where $l < d$. Now consider the matrix

$$M = \begin{pmatrix} \alpha^{jm_1} & \alpha^{jm_2} & \dots & \alpha^{jm_l} \\ \alpha^{(j+1)*m_1} & \alpha^{(j+1)*m_2} & \dots & \alpha^{(j+1)*m_l} \\ \vdots & \vdots & & \vdots \\ \alpha^{(j+l-1)*m_1} & \alpha^{(j+l-1)*m_2} & \dots & \alpha^{(j+l-1)*m_l} \end{pmatrix}$$

and the vector v containing the coefficients of the polynomial v. Since the $\alpha^j \dots \alpha^{j+d-2}$ are roots of v, we have $Mv = 0$. We know that all columns of M contains a geometric progression, consequently $Det(M)$ is a nonzero scalar multiplied by a Vandermonde determinant with generators $\alpha^j * m_i$. But since $Ord_* \alpha \geq l$, these generators are pairwise distinct, consequently $Det(M)$ is nonzero, meaning that $Mv = 0$ implies $v = 0$. This contradiction proves our initial point.■

2.9 Cyclic Codes

To understand cyclic codes I relied on [7].

2.9.1 Cyclic shift

Definition Let

$$v = (v_0, v_1, \dots, v_{n-1})$$

denote an n-tuple. Te operation

$$(v_0, v_1, \dots, v_{n-1})^{(1)} = (v_{n-1}, v_0, \dots v_{n-2})$$

is called the **cyclic shift of** v. Let us denote

$$(v_0, v_1, \dots, v_{n-1})^{(\Psi)} = (v_{n-\Psi}, v_{n-\Psi+1}, \dots v_{n-\Psi-1})$$

which is shifting Ψ places to the right. The indices are to be understood modulo n.

Indeed, it comes naturally that

$$v^{(\Psi_1)} = v^{(\Psi_2)} \leftarrow \Psi_1 \equiv \Psi_2 \pmod{n}$$

The converse does not necessarily hold, since for instance in the case of

$$e_3 = (1,1,1)$$

$$e_3^{(1)} = e_3^{(2)} = (1,1,1) = e_3$$

but

$$\neg(1 \equiv 2 \pmod{3})$$

is true. One can easily see that shifting a vector Ψ places to the right is equivalent with shifting it $n-\Psi$ places to the left. For example

$$e_n = (1, 1, \dots 1)$$

is a fixed point of this operator for all n. One can observe without difficulties that

$$\forall v : v^{(0)} = \mathrm{id}(v) = v$$

2.9.2 Connection between cyclic codes and polynomials

Definition Let C be a code. We say that C is cyclic by definition if and only if C is closed under the operation $v^{(\Psi)}$ meaning that

$$\forall c \in C \ \forall \Psi \in \mathbb{Z} \ c^{(\Psi)} \in C$$

holds as well.

Indeed, we will represent these codes in their polynomial form so that we can comprehend them more easily and we can perform operations more conveniently. The polynomial corresponting to v is

$$v(X) = \sum_{\gamma=0}^{n-1} v_\gamma X^\gamma$$

and the polynomial which corresponds to $v^{(\Psi)}$ is

$$v^{(\Psi)}(X) = \sum_{\gamma=0}^{n-1} v_{n-\Psi+\gamma} X^\gamma.$$

It is fairly obvious that

$$X^\Psi v(X) = X^\Psi \sum_{\gamma=0}^{n-1} v_\gamma X^\gamma = \sum_{\gamma=0}^{n-1} v_\gamma X^\gamma X^\Psi = \sum_{\gamma=0}^{n-1} v_\gamma X^{\gamma+\Psi}$$

holds. We can easily rewrite

$$X^\Psi v(X) = q(X)(X^n + 1) + v^{(i)}(X)$$

in the binary case, and when discussing generaly

$$X^\Psi v(X) = q(X)(X^n - 1) + v^{(i)}(X).$$

In the following part of this subsection, we will only be concerned about binary codes.

Theorem 2.32 *A polynomial g generates a cyclic code if and only if*

$$g \equiv 0 \ (mod \ X^n + 1)$$

holds.

2.9.3 Examples, connection with repetition codes

The following example will help one comprehend the concept. We do indeed know that $g(X) = 1 + X + X^2$ generates a cyclic code, since

$$g(X) = 1 + X + X^2 \equiv 0 \pmod{X^3 + 1}$$

because

$$X^3 + 1 = (1 + X)(X^2 + X + 1).$$

We simply get

$$0 \cdot (X^2 + X + 1) = 0$$
$$1 \cdot (X^2 + X + 1) = X^2 + X + 1$$

meaning that

$$\phi(0) = (000)$$
$$\phi(1) = (111)$$

which is obviously the repetition code of length 3 denoted by $RC(3)$. We can simply observe that

$$g(x) = \prod_{\gamma=0}^{n-1} X^\gamma = 1 + X + \cdots + X^{n-1}$$

generates a cyclic code as well. For this it is sufficient to see the binary version of the identity well-known from high school

$$X^n + 1 = (1 + X) \prod_{\gamma=0}^{n-1} X^\gamma = (1 + X)(1 + X + \cdots + X^{n-1})$$

implying that

$$g(x) = \prod_{\gamma=0}^{n-1} X^\gamma = 1 + X + \cdots + X^{n-1} \equiv 0 \pmod{X^n + 1}.$$

Obviously

$$0 \cdot \prod_{\gamma=0}^{n-1} X^\gamma = 0$$
$$1 \cdot \prod_{\gamma=0}^{n-1} X^\gamma = \prod_{\gamma=0}^{n-1} X^\gamma$$

therefore

$$\phi(0) = (00 \ldots 0)$$
$$\phi(1) = (11 \ldots 1)$$

meaning that we obtained the repetition code of length n denoted by $RC(n)$.

For our third example et us consider the polinomial

$$g(X) = 1 + X + X^3.$$

We know that

$$X^7 + 1 = (1 + X)(1 + X + X^3)(1 + X^2 + X^3)$$

consequently

$$g(X) = 1 + X + X^3 \equiv 0 \pmod{X^7 + 1}$$

is true, therefore $g(X)$ generates a cyclic code. Obviously

$$0 \cdot (1+X+X^3) = 0$$

$$1 \cdot (1+X+X^3) = 1+X+X^3$$
$$X \cdot (1+X+X^3) = X+X^2+X^4$$
$$(1+X) \cdot (1+X+X^3) = 1+X^2+X^3+X^4$$
$$X^2 \cdot (1+X+X^3) = X^2+X^3+X^5$$
$$(1+X^2) \cdot (1+X+X^3) = 1+X+X^2+X^5$$
$$(X+X^2) \cdot (1+X+X^3) = X+X^3+X^4+X^5$$
$$(1+X+X^2) \cdot (1+X+X^3) = 1+X^4+X^5$$
$$X^3(1+X+X^3) = X^3+X^4+X^6$$
$$(1+X^3) \cdot (1+X+X^3) = 1+X+X^4+X^6$$
$$(X+X^3) \cdot (1+X+X^3) = X+X^2+X^3+X^6$$
$$(1+X+X^3) \cdot (1+X+X^3) = 1+X^2+X^6$$
$$(X^2+X^3) \cdot (1+X+X^3) = X^2+X^4+X^5+X^6$$
$$(1+X^2+X^3) \cdot (1+X+X^3) = 1+X+X^2+X^3+X^4+X^5+X^6$$
$$(X+X^2+X^3) \cdot (1+X+X^3) = X+X^5+X^6$$
$$(1+X+X^2+X^3) \cdot (1+X+X^3) = 1+X^3+X^5+X^6$$

therefore the generated codewords with the original message are the following

$$\phi(0000) = (0000000)$$

$$\phi(1000) = (1101000)$$
$$\phi(0100) = (0110100)$$
$$\phi(1100) = (1011100)$$
$$\phi(0010) = (0011010)$$
$$\phi(1010) = (1110010)$$
$$\phi(0110) = (0101110)$$
$$\phi(1110) = (1000110)$$
$$\phi(0001) = (0001101)$$
$$\phi(1001) = (1100101)$$
$$\phi(0101) = (0111001)$$
$$\phi(1101) = (1010001)$$
$$\phi(0011) = (0010111)$$
$$\phi(1011) = (1111111)$$
$$\phi(0111) = (0100011)$$
$$\phi(1111) = (1001011).$$

For the message $(abcd)$ we multiplied $g(X)$ with $a+bX+cX^2+dX^3$ and obtained the codeword.

3 Reed-Muller Codes

3.1 Recursive definition, examples, some elementary attributes

3.1.1 Recursive definition and examples for x=1

This subsection utilizes [14],[16], [17] and [18].

Reed-Muller codes can be defined in various ways, for example by their generator matrices. We will utilize the recursive way for now. Later a plethora of other ways to define/understand Reed-Muller codes will be detailed in this work. Let us note that we will only be elaborating on the concept of binary Reed-Muller codes, and whenever we mention Reed-Muller codes, we want to refer to binary Reed-Muller codes. Otherwise we will say "Generalized Reed-Muller codes".

Definition For $y \geq 0$ let

$$RM(0,y) = RC(2^y)$$

be the the repetition code with length 2^y. These codes are called the **zeroth orrder Reed-Muller codes**. The **first order Reed-Muller codes** are defined in two steps. First we have

$$RM(1,1) = \mathbb{F}_2^2,$$

and second we have

$$RM(1,y+1) = \{(u,u) \mid u \in RM(1,y)\} \cup \{(u,u+e) \mid u \in RM(1,y)\}$$

for $y \geq 1$, where e denotes a vector whose every coordinate is 1. The higher order Reed-Muller codes are defined by the recursive formulae

$$RM(x,x) = \mathbb{F}_2^{2^x}$$

if x is at least 2 and

$$RM(x,y) = \{(u,u+v) \mid (u \in RM(x,y-1)) \wedge (v \in RM(x-1,y-1))\}$$

if x and y are different and $x, y \geq 2$. The code $RM(x,y)$ is by definition said to be a **Reed-Muller code of order x**.

Let us examine how the most basic Reed-Muller codes really look like. It follows immediately from the definition that

$$RM(1,1) = \{00, 01, 10, 11\}.$$

To obtain $RM(1,2)$ we need to concatenate codewords of $RM(1,1)$ with themselves and concatenate every codeword with the version of itself added to the vector whose every coordinate is 1. Consequently it follows that

$$RM(1,2) = \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}$$

holds. By now one has a clear idea about that the fact that the number of codewords doubles in every iteration concerning the first order Reed-Muller codes.

$$RM(1,3) = \left\{\begin{matrix} 00000000, 01010101, 10101010, 11111111 \\ 00110011, 01100110, 10011001, 11001100 \\ 00001111, 01011010, 10100101, 11110000 \\ 00111100, 01101001, 10010110, 11000011 \end{matrix}\right\}$$

Next example of first order Reed-Muller codes is

$$RM(1,4) = \begin{Bmatrix} 0000000000000000, 0101010101010101 \\ 1010101010101010, 1111111111111111 \\ 0011001100110011, 0110011001100110 \\ 1001100110011001, 1100110011001100 \\ 0000111100001111, 0101101001011010 \\ 1010010110100101, 1111000011110000 \\ 0011110000111100, 0110100101101001 \\ 1001011100101100, 1100001111000011 \\ 0000000011111111, 0101010110101010 \\ 1010101001010101, 1111111100000000 \\ 0011001111001100, 0110011010011001 \\ 1001100101100110, 1100110000110011 \\ 0000111111110000, 0101101010100101 \\ 1010010101011010, 1111000000001111 \\ 0011110011000011, 0110100110010110 \\ 1001011001101001, 1100001100111100 \end{Bmatrix}$$

from where we can start to get an impression how first order codes really operate. Our last example has a high significance, since the code $RM(1,5)$ has been utilized by **Mariner 9** to send back black and white pictures from the surface of Mars. The code has been selected because of the fact that it can be decoded really rapidely. More than 7000 pictures has been sent back to Earth and approximately 85 percent of the surface of Mars has been mapped. We note that the year of the successful mission was 1971.

$$RM(1,5)=\left\{\begin{array}{c}
00000000000000000000000000000000\\
01010101010101010101010101010101\\
10101010101010101010101010101010\\
11111111111111111111111111111111\\
00110011001100110011001100110011\\
01100110011001100110011001100110\\
10011001100110011001100110011001\\
11001100110011001100110011001100\\
00001111000011110000111100001111\\
01011010010110100101101001011010\\
10100101101001011010010110100101\\
11110000111100001111000011110000\\
00111100001111000011110000111100\\
01101001011010010110100101101001\\
10010111001011001001011100101100\\
11000011110000111100001111000011\\
00000000111111111111111100000000\\
01010101101010101010101001010101\\
10101010010101010101010110101010\\
11111111000000000000000011111111\\
00110011110011001100110000110011\\
01100110100110011001100101100110\\
10011001011001100110011010011001\\
11001100001100110011001111001100\\
00001111111100001111000000001111\\
01011010101001011010010101011010\\
10100101010110100101101010100101\\
11110000000011110000111111110000\\
00111100110000111100001100111100\\
01101001100101101001011001101001\\
10010110011010010110100110010110\\
11000011001111000011110011000011
\end{array}\right\}$$

3.1.2 Parameters of Reed-Muller codes for x=1, weight of codewords

The next theorem will highlight some elementary attributes of first order Reed-Muller codes.

Theorem 3.1 *Let y be at least* 1*.*

1. $RM(1,y)=[2^y, y+1, 2^{y-1}]_2$.
2. *The Reed-Muller code $RM(1,y)$ contains only even weighted words with length 2^y.*
3. *Every codeword in $RM(1,y)$ except from* 0 *and e has a weight 2^{y-1}.*

3.1.3 Higher order Reed-Muller codes

Now we will see two examples of higher order Reed-Muller codes. Let us see an instance where x and y are the same. It is reasonable to select a small number, since the cardinality of the code will be of course 2^{2^x}. By the definition we obtain that

$$RM(2,2) = \mathbb{F}_2^{2^2} = \mathbb{F}_2^4 = \begin{Bmatrix} 0000, 0001, 0010, 0011 \\ 0100, 0101, 0110, 0111 \\ 1000, 1001, 1010, 1011 \\ 1100, 1101, 1110, 1111 \end{Bmatrix}$$

holds. If we want to determine how $RM(2,3)$ looks like, we have to concatenate the codeswords of $RM(2,2)$ with the codewords of $RM(2,2)$ added to the codewords of $RM(1,2)$.

$$RM(2,3) = \begin{Bmatrix} 00000000, 00000101, 00001010, 00001111 \\ 00000011, 00000110, 00001001, 00001100 \\ 00010001, 00010100, 00011011, 00011110 \\ 00010010, 00010111, 00011000, 00011101 \\ 00100010, 00100111, 00101000, 00101101 \\ 00100001, 00100100, 00101011, 00101110 \\ 00110011, 00110110, 00111001, 00111100 \\ 00110000, 00110101, 00111010, 00111111 \\ 01000100.01000001, 01001110, 01001011 \\ 01000111, 01000010, 01001101, 01001000 \\ 01010101, 01010000, 01011111, 01011010 \\ 01010110, 01010011, 01011100, 01011001 \\ 01100110, 01100011, 01101100, 01101001 \\ 01100101, 01100000, 01101111, 01101010 \\ 01110111, 01110010, 01111101, 01111000 \\ 01110100, 01110001, 01111110, 01111011 \\ 10001000, 10001101, 10000010, 10000111 \\ 10001011, 10001110, 10000001, 10000100 \\ 10011001, 10011100, 10010011, 10010110 \\ 10011010, 10011111, 10010000, 10010101 \\ 10101010, 10101111, 10100000, 10100101 \\ 10101001, 10101100, 10100011, 10100110 \\ 10111011, 10111110, 10110001, 10110100 \\ 10111000, 10111101, 10110001, 10110111 \\ 11001100, 11001001, 11000110, 11000011 \\ 11001111, 11001010, 11000101, 11000000 \\ 11011101, 11011000, 11010111, 11010010 \\ 11011110, 11011011, 11010100, 11010001 \\ 11101110, 11101011, 11100100, 11100001 \\ 11101101, 11101000, 11100111, 11100010 \\ 11111111, 11111010, 11110101, 11110000 \\ 11111100, 11111001, 11110110, 11110011 \end{Bmatrix}$$

The code $RM(3,3)$ would contain $2^{2^3} = 2^8 = 256$ codewords.

3.1.4 Parameters of higher order Reed-Muller codes and connection with binomial coeffiicients

The $RM(2,3)$ was our last example of higher order Reed-Muller codes. Now we will explore some of the easier features of the Reed-Muller codes in general.

Theorem 3.2 *For all x and y we have*

$$RM(x,y) = \left[2^y, \sum_{i=0}^{x} \binom{y}{i}, 2^{y-x}\right]_2.$$

Proof 1. It immediately follows from 2.28.

2. We can see from 2.28 that k is additive. Considering the identities

$$\binom{y}{0} = \binom{y-1}{0} = 1$$

and

$$\binom{y-1}{i-1} + \binom{y-1}{i} = \binom{y}{i}.$$

it can be obtained that

$$\sum_{i=0}^{x} \binom{y-1}{i} + \sum_{i=0}^{x-1} \binom{y-1}{i} = \sum_{i=0}^{x} \binom{y}{i},$$

proving what we initially wanted.

3. Again by utilizing 2.28 we get that

$$d(RM(x,y)) = min(2d(RM(x,y-1)), d(RM(x-1,y-1))).$$

Induction will show us that

$$d(RM(x,y)) = min(2 * 2^{y-1-x}, 2^{y-1-(x-1)}) = min(2^{y-x}, 2^{y-x}) = 2^{y-x}.$$

■

3.2 Generator matrices and duals of Reed-Muller codes

I relied only on [17] and [18] in this subsection.

3.2.1 Recursive construction of generator matrices

From the recursive definition of Reed-Muller codes the generator matrices come really easily. If the matrices $G_{RM(x,y-1)}$ and $G_{RM(x-1,y-1)}$ are defined, we have

$$G_{RM(x,y)} = \begin{pmatrix} G_{RM(x,y-1)} & 0 \\ G_{RM(x,y-1)} & G_{RM(x-1,y-1)} \end{pmatrix}.$$

The other cases are fairly easy as well. If $G_{RM(1,y-1)}$ is defined, we have

$$G_{RM(1,y)} = \begin{pmatrix} G_{RM(1,y-1)} & 0 \\ G_{RM(1,y-1)} & 1\dots1 \end{pmatrix}.$$

Furthermore

$$G_{RM(1,1)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

3.2.2 Duals of Reed-Muller codes

Theorem 3.3 *1. The Reed-Muller codes $RM(x,y)$ and $RM(y-x-1,y)$ are dual codes of each other.*

2. The Reed-Muller code $RM(x,y)$ is self dual if $y=2x-1$.

3. The converse of these also hold, namely $RM(x_1,y_1)$ and $RM(x_2,y_2)$ are duals if and only if $y_2=y_1$ and $x_2=y_1-x_1-1$, furthermore $RM(x,y)$ is self dual if and only if $y=2x-1$.

Proof The second claim follows from the first one immediately. The first claim will be proven by induction. Let $y=2$. In this case the codes $RM(x,y)$ and $RM(y-x-1,y)$ both exists if and only if $x=0$ or $x=1$. If $x=0$ and $y=2$ our two codes are $RM(0,2)$ and $RM(1,2)$. If $x=1$ and $y=2$ the two codes in question are $RM(0,2)$ and $RM(1,2)$, therefore the two cases give us the same pair of codes. We know that

$$RM(0,2)=\{0000,1111\}$$

and

$$RM(1,2)=\{0000,0101,1010,1111,0011,0110,1001,1100\}$$

from the definition. Since in $RM(1,2)$ all codewords have even weight and $RM(0,2)$ is the repetition code with length of codewords 4, they will be orthogonal to each other. Because of 2.26 we only need to show that

$$G^T_{RM(x,y)} * G_{RM(x-y-1,y)} = 0,$$

meaning that the columns of $G_{RM(x,y)}$ are orthogonal to the columns of $G_{RM(x-y-1,y)}$, and that

$$Rank(G_{RM(x,y)}) + Rank(G_{RM(x-y-1,y)}) = y.$$

We will utilize induction by y. We assume that the theorem holds to $y-1$. The orthogonality of the columns comes from the induction hypothesis and the claim concerning the rank is derived from the identity

$$\sum_{i=0}^{x}\binom{y}{i}+\sum_{i=0}^{y-x-1}\binom{y}{i}=\sum_{i=0}^{x}\binom{y}{i}+\sum_{i=x+1}^{y}\binom{y}{i}=2^y.$$

In the case of the third claim we only have to consider that the dual of a code is unique and that the dual of $RM(x_1,y_1)$ is $RM(y_1-x_1-1,y_1)$. ■

3.3 Reed Decoding

The cource I claim to utilize to present Reed Decoding is [17].

In this section we will show an algorithm to decode Reed-Muller codes with the involvement of simple elementary tools. This particular algorithm is called the **Reed Decoding**. We will show the algorithm in practice by the example of $RM(1,3)$. A possible generator matrix of $RM(1,3)$ is

$$G_{RM(1,3)}=\begin{pmatrix}1&0&0&0\\1&1&0&0\\1&0&1&0\\1&1&1&0\\1&0&0&1\\1&1&0&1\\1&0&1&1\\1&1&1&1\end{pmatrix}.$$

The columns of $G_{RM(1,3)}$ form a basis of $RM(1,3)$, let us call them v_i. For any codeword $v \in RM(1,3)$ there is a tuple (a_0, a_1, a_2, a_3) such as

$$v = a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3$$

Now we obtain

$$v = \begin{pmatrix} a_0 \\ a_0 + a_1 \\ a_0 + a_2 \\ a_0 + a_1 + a_2 \\ a_0 + a_3 \\ a_0 + a_1 + a_3 \\ a_0 + a_2 + a_3 \\ a_0 + a_1 + a_2 + a_3 \end{pmatrix}.$$

Let us assume, that the vector we receive is

$$w = \begin{pmatrix} w_0 & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 \end{pmatrix}.$$

Let us examine first the case when no error occurs. In this case the equations

1. $a_0 = w_0$
2. $a_1 = w_0 + w_1 = y_2 + w_3 = w_4 + w_5 = w_6 + w_7$
3. $a_2 = w_0 + w_2 = w_1 + w_3 = y_4 + w_6 = w_5 + w_7$
4. $a_3 = w_0 + w_4 = w_1 + w_5 = w_2 + w_6 = w_3 + w_7$

hold, and we can easily determine what the intended message really is. Now assume that we have exactly one error. In this case in the last three lines exactly one value will be different from the others. If we ignore those values and solve the remaining equations, we can easily decode the message. We can aslo easily obtain, that a_0 is the most frequent component of

$$Q = w + a_1 v_1 + a_2 v_2 + a_3 v3 = \begin{pmatrix} w_0 \\ w_1 + a_1 \\ w_2 + a_2 \\ w_3 + a_1 + a_2 \\ w_4 + a_3 \\ w_5 + a_1 + a_3 \\ w_6 + a_2 + a_3 \\ w_7 + a_1 + a_2 + a_3 \end{pmatrix}.$$

First let us assume that the error is in w_0, meaning that $w_0 \neq a_0$, and the other characters are intact. In this case We can easily obtain that

$$Q = \begin{pmatrix} y_0 \\ a_0 + a_1 + a_1 \\ a_0 + a_2 + a_2 \\ a_0 + a_1 + a_2 + a_1 + a_2 \\ a_0 + a_3 + a_3 \\ a_0 + a_1 + a_3 + a_1 + a_3 \\ a_0 + a_2 + a_3 + a_2 + a_3 \\ a_0 + a_1 + a_2 + a_3 + a_1 + a_2 + a_3 \end{pmatrix} = \begin{pmatrix} w_0 \\ a_0 \\ a_0 \\ a_0 \\ a_0 \\ a_0 \\ a_0 \\ a_0 \end{pmatrix},$$

which means that the most frequent coordinate will be a_0. Now following this example we can show that if the error is in w_i, then row i will not give a_0, but all the others will. Let us observe how the algorithm operates in a simple case. Say that the codeword $v^T = (1,1,1,1,1,1,1,1)$ is received as $w^T = (0,1,1,1,1,1,1,1)$. Considering 3.3 it is obtained that

1. $a_1 = 1 = 0 = 0 = 0$

2. $a_2 = 1 = 0 = 0 = 0$

3. $a_3 = 1 = 0 = 0 = 0.$

We will omit those values which do not agree with the others and we can infer that $a_1, a_2, a_3 = 0$. Since

$$Q = w + a_1v_1 + a_2v_2 + a_3v3 = \begin{pmatrix} 0 \\ 1+0 \\ 1+0 \\ 1+0+0 \\ 1+0 \\ 1+0+0 \\ 1+0+0 \\ 1+0+0+0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

we know that the error was in w_0. It can be obtained without difficulties that

$$v = a_0v_0 + a_1v_1 + a_2v_2 + a_3v_3 = v_0 = (1111111)^T.$$

3.4 Decoding with Hadamard Matrices, relationship with Kronecker Product

For the contemplation of the connection between Hadamard matrices and Reed-Muller codes I rely on [15], [18] and [19].

3.4.1 Introduction to Kronecker product

Definition Let A and B be matrices. The **Kronecker product** of A and B is defined as

$$A \otimes B = (a_{ij}B)_{kl}.$$

More explicitly we can write

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1q}B \\ a_{21}B & a_{22}B & \dots & a_{2q}B \\ \vdots & \dots & \vdots & \dots \\ a_{p1}B & a_{p2}B & \dots & a_{pq}B \end{pmatrix}$$

or even more explicitly

$$A \otimes B = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & \dots & a_{11}b_{1s} & a_{12}b_{11} & a_{12}b_{12} \dots a_{1q}b_{1s} \\ a_{11}b_{21} & a_{11}b_{22} & \dots & a_{11}b_{2s} & a_{12}b_{11} & a_{12}b_{22} \dots a_{1q}b_{2s} \\ \vdots & \dots & \vdots & \dots & \dots & \vdots \\ a_{11}b_{r1} & a_{11}b_{r2} & \dots & a_{11}b_{rs} & a_{12}b_{r1} & a_{12}b_{r2} \dots a_{1q}b_{rs} \\ \vdots & \dots & \vdots & \dots & \dots & \vdots \\ a_{p1}b_{(r-1)1} & a_{p1}b_{(r-1)2} & \dots & a_{p1}b_{(r-1)s} & a_{p2}b_{(r-1)1} & a_{p2}b_{(r-1)2} \dots a_{pq}b_{(r-1)s} \\ a_{p1}b_{r1} & a_{p1}b_{r2} & \dots & a_{p1}b_{rs} & a_{p2}b_{11} & a_{p2}b_{12} \dots a_{pq}b_{rs} \end{pmatrix}.$$

It comes naturally from the definition that the Kronecker product is not a commutative operation. For instance consider

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \\ 1 & 2 & 1 & 2 \\ 3 & 4 & 3 & 4 \end{pmatrix}$$

and

$$\begin{pmatrix}1 & 2\\3 & 4\end{pmatrix} \otimes \begin{pmatrix}1 & 1\\1 & 1\end{pmatrix} = \begin{pmatrix}1 & 1 & 2 & 2\\1 & 1 & 2 & 2\\3 & 3 & 4 & 4\\3 & 3 & 4 & 4\end{pmatrix}$$

as a simple counter-example. One can easily consider that for all A matrices

$$E_1 \otimes A = A = A \otimes E_1$$

holds. Furthermore if 0_A denotes the zero matrix which has exactly the size of matrix A then

$$0_{E_1} \otimes A = 0_A = A \otimes 0_{E_1}$$

follows as well. The Kronecker product is bilinear, since

$$\Psi(A \otimes B) = \Psi((a_{ij}B)_{kl}) = (\Psi(a_{ij}B))_{kl} = ((\Psi a_{ij}B))_{kl} = (\psi A) \otimes B$$

$$\Psi(A \otimes B) = \Psi((a_{ij}B)_{kl}) = (\Psi(a_{ij}B))_{kl} = (a_{ij}(\Psi B))_{kl} = A \otimes (\Psi B)$$

furthermore

$$A \otimes (B + C) = (a_{ij}(B + C))_{kl} = (a_{ij}B + a_{ij}C)_{kl} =$$
$$= (a_{ij}B)_{kl} + (a_{ij}C)_{kl} = A \otimes B + A \otimes C$$

for the right variable, and

$$(A + B) \otimes C = ((A + B)_{ij}C)_{kl} = ((a_{ij} + b_{ij})C)_{kl} =$$
$$= (a_{ij}C + b_{ij}C)_{kl} = (a_{ij}C)_{kl} + (b_{ij}C)_{kl} = A \otimes C + B \otimes C$$

for the left variable.

Indeed, the associativity of the operation can also be discovered fairly easily merely from the definition and raw calculation.

$$(A \otimes B) \otimes C = (a_{ij}B)_{kl} \otimes C = (a_{ij}b_{kl}C)_{mn}$$

$$A \otimes (B \otimes C) = A \otimes ((b_{kl}C)_{mn}) = (a_{ij}b_{kl}C)_{mn}$$

Additionally, the Kronecker product is a special case of the tensor product, and has a plethora of interesting properties. We now would like to explore how it contributes to decoding Reed-Muller codes.

3.4.2 Kronecker product for more variables

For our decoding algorithm we will need to build large Hadamard matrices 3.4.3, so we need to generalize the kronecker product for more variables 3.4.5.

Definition For the **empty Kronecker product**

$$\otimes_{i \in \emptyset} M_i = E_1 = \begin{pmatrix}1\end{pmatrix}.$$

Let $I \subseteq \mathbb{N}$ be a set of indices so that $\mathrm{Card}(I) \in \mathbb{N}^+$. Let $(M_i)_{i \in I}$ be a sequence of matrices. Let

$$j = \mathrm{Max}(I).$$

The **multivariable Kronecker product** is defined as

$$\otimes_{i \in I} = (\otimes_{i \in I \setminus \{j\}} M_i) \otimes M_j.$$

For instance if $I = \{1\}$ and (M_1) is a sequence containing only one matrix, then

$$\otimes_{i\in\{1\}} M_i = (\otimes_{i\in\{1\}\setminus\{1\}} M_i) \otimes M_1 = (\otimes_{i\in\emptyset} M_i) \otimes M_1 = \begin{pmatrix}1\end{pmatrix} \otimes M_1 = M_1$$

If $I = \{1, 2\}$ and our matrices are (M_1, M_2), then

$$\otimes_{i\in\{1,2\}} M_i = (\otimes_{i\in\{1\}} M_i) \otimes M_2 = M_1 \otimes M_2.$$

For $I = \{1, 2, 3\}$ we have

$$\otimes_{i\in\{1,2,3\}} M_i = (\otimes_{i\in\{1,2\}} M_i) \otimes M_3 = (M_1 \otimes M_2) \otimes M_3 = M_1 \otimes (M_2 \otimes M_3) = M_1 \otimes M_2 \otimes M_3.$$

Because of the associativity of the Kronecker product, there is no need to write parentheses. We can also write

$$\otimes_{i\in\{1,2,\ldots n\}} M_i = \otimes_{i=1}^{n} M_i = M_1 \otimes M_2 \otimes \cdots \otimes M_n$$

3.4.3 Hadamard matrices

Definition Let H_n be an $n \times n$ matrix whose all elements are either 1 or -1, and moreover

$$H_n H_n^T = nE_n.$$

These matrices are by definition called **Hadamard matrices of order n**.

$$H_1 = \begin{pmatrix}1\end{pmatrix}.$$

$$H_2 = \begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}.$$

$$H_4 = H_2 \otimes H_2 = \begin{pmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ 1 & 1 & -1 & -1\\ 1 & -1 & -1 & 1\end{pmatrix}.$$

It is clear from the definition that $-H_n$ is also a Hadamard matrix, since

$$(-H_n)((-H_n)^T) = (-H_n)(-H_n^T) = H_n H_n^T = nE_n.$$

Therefore for instance the matrices

$$\widehat{H_1} = \begin{pmatrix}-1\end{pmatrix}.$$

$$\widehat{H_2} = \begin{pmatrix}-1 & -1\\ -1 & 1\end{pmatrix}.$$

$$\widehat{H_4} = \begin{pmatrix}-1 & -1 & -1 & -1\\ -1 & 1 & -1 & 1\\ -1 & -1 & 1 & 1\\ -1 & 1 & 1 & -1\end{pmatrix}.$$

are Hadamard matrices as well. Furthermore, the matrix obtained by the permutation of rows and columns is also a Hadamard matrix, consequently for instance

$$\widehat{\widehat{H_2}} = \begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}.$$

is a Hadamard matrix as well. We also have that if H_n is a Hadamard matrix then H_n^T is a Hadamard matrix as well, since

$$H_n H_n^T = nE_n \leftrightarrow H_n^T H_n = nE_n \leftrightarrow H_n^T (H_n^T)^T = nE_n$$

therefore for instance

$$\widehat{\widehat{\widehat{H}}}_2 = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

is a Hadamard matrix as well.

It is also not difficult to consider that

$$\mathrm{Det}(H_n) = \pm n^{\frac{n}{2}}$$

since

$$H_n H_n^T = nE_n \to \mathrm{Det}(H_n H_n^T) = \mathrm{Det}(nE_n) \leftrightarrow \mathrm{Det}(H_n)\mathrm{Det}(H_n^T) = n^n \mathrm{Det}(E_n) \leftrightarrow$$
$$\leftrightarrow \mathrm{Det}(H_n)\mathrm{Det}(H_n) = n^n \leftrightarrow \mathrm{Det}(H_n)^2 = n^n \leftrightarrow \mathrm{Det}(H_n) = \pm\sqrt{n^n} \leftrightarrow \mathrm{Det}(H_n) = \pm n^{\frac{n}{2}}.$$

Definition The Hadamard matrices H and H' are by definition said to be **equivalent** if and only H' can be obtained from H by permutating rows or columns or by multiplying by -1. The Hadamard matrix H is said to be **normalized** by definition if and only if the first row of H and the first column of H only contains 1.

It comes without difficulties to deduce that the equivalence of Hadamard matrices is an equivalence relation. It is obvious that evey Hadamard matrix is equivalent with a normalized Hadamard matrix. The algorithm under consideration will decode $RM(1, y)$ and will involve H_{2^y}.

3.4.4 Size of Hadamard matrices

The following theorem with its proof will highlight and summarize some of the most elementary attributes of Hadamard matrices.

Theorem 3.4 *Let H_n be a Hadamard matrix of order n. We have $n \leq 2$ or $n \equiv 0$ $(mod\ 4)$.*

Proof First we have to obtain that the order of a Hadamard matrix cannot be 3. This case can be easily done by manually checking all possibilities. This comes without difficulties and huge effort, since if H_n is not a Hadamard matrix, we do not have to check $-H_n$ and all the H_n' matrices derived from H_n by permutating rows and columns and we do not have to check H_n^T either. Let now $n \geq 4$. We can assume without the loss of generality that H_n is in normalized form. The first and the second row is orthogonal to each other and the first only contains ones, the second one contains $n/2$ ones and $n/2$ minus ones. We can assume without the loss of generality that the first $n/2$ elements are 1, and the last $n/2$ are -1. Let us now examine row 3. Let us denote with a the number of 1's where there is an 1 in the first and the second row above, let b denote the number of -1's, where there is a 1 in the first and the second row above, let c be the number of 1's where there is a 1 in the first and a -1 in the second row above, and let d be the number of -1's where there is a 1 in the first and -1 in the second row above. We know that $a + b = n/2$ and $c + d = n/2$. Now we calculate the scalar product of the first and the third row, and it is obtained that

$$a - b + c - d = 0.$$

If we calculate the inner product of the second and the third row we get

$$a - b - c + d = 0.$$

The obtained system is really easy to solve, we can infer that

$$a = b = c = d = n/4$$

which implies that $n \equiv 0$ (mod 4).■

Note that from this theorem we can conclude that the determinant of Hadamard matrices is always a natrual number

$$\mathrm{Det}(H_n) = \pm n^{\frac{n}{2}} \in \mathbb{N}$$

and for $n \geq 2$ we have

$$\mathrm{Det}(H_{4k}) = \pm(4k)^{\frac{4k}{2}} = \pm(4k)^{2k} = \pm 16^k \cdot k^{2k}$$

for every Hadamard matrix.

3.4.5 Construction of larger Hadamard matrices

Since for the decoding algorithm we need H_{2^y}, we need an algorithm to construct it rapidly.

Theorem 3.5 *Let I be a finite index set. Let H_i Hadamard matrices, where $i \in I$. We have*

$$\otimes_{i \in I} H_{n_i} = H_{\prod_{i \in I} n_i}.$$

In other words the product of Hammad matrices is also a Hadamard matrix, and the order will be the product of the orders.

Proof It is sufficient to show the $Card(I) = 2$ case, the other cases can be proven by induction on $Card(I)$. First we will prove that

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

We have

$$(A \otimes B)(C \otimes B) = (a_{ij}B)_{ij}(c_{kl}D)_{kl} = (a_{ij}c_{kl}BD)$$

and

$$(AC) \otimes (BD) = (a_{ij}c_{kl}) \otimes (BD) = (a_{ij}c_{kl}BD)$$

by the definition of Kronecker product and matrix multiplication. It is also clear that

$$(A \otimes B)^T = A^T \otimes B^T$$

since

$$(A \otimes B)^T = (a_{ij}B)^T_{ij} = (a_{ji}B^T)_{ji}$$

and

$$A^T \otimes B^T = (a_j i)_{ji} \otimes B^T = (a_{ji}B^T)_{ji}.$$

One can see without difficulties that $E_m \otimes E_n = E_{mn}$ and consequently

$$(mE_m) \otimes (nE_n) = mnE_{mn}.$$

Let $A = C = H_m$ and $B = D = H_n$. It is obtained that

$$(H_m \otimes H_n)(H_m \otimes H_n)^T = (H_m \otimes H_n)(H_m^T \otimes H_n^T) =$$

$$= (H_m H_m^T) \otimes (H_n H_n^T) = (mE_m) \otimes (nE_n) = mnE_{mn}$$

completing the proof.■

Theorem 3.6 *Let*

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Construct the matrices

$$M_{2^y}^{(i)} = E_{2^{y-i}} \otimes H_2 \otimes E_{2^{i-1}}$$

for $1 \le i \le y$. We have

$$H_{2^y} = \prod_{i=1}^{y} M_{2^y}^{(i)} = M_{2^y}^{(1)} M_{2^y}^{(2)} \dots M_{2^y}^{(y)}.$$

In other words if we multiply the matrices constructed above, we obtain a Hadamard matrix of order 2^y.

Proof We will utilize induction by y. First let us examine the $y = 1$ case. We have only the matrix

$$M_2^1 = M_{2^1}^{(1)} = E_{2^0} \otimes H_2 \otimes E_{2^0} = E_1 \otimes H_2 \otimes E_1 = H_2 \otimes E_1 = H_2$$

and

$$H_2 = H_{2^1} = \prod_{i=1}^{1} M_{2^1}^{(i)} = M_2^1$$

Now let $y > 1$. It is easily obtained that for $0 \le i \le y$

$$M_{2^{y+1}}^{(i)} = E_{2^{y+1-i}} \otimes H_2 \otimes E_{2^{i-1}} = E_2 \otimes E_{2^{y-i}} \otimes H_2 \otimes E_{2^{i-1}} = E_2 \otimes M_{2^y}^{(i)}.$$

It is also clear that

$$M_{2^{y+1}}^{y+1} = H_2 \otimes E_{2^y}.$$

We have already seen the formula

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD).$$

We now need the generalized version of this, namely

$$\prod_{i \in I} (A_i \otimes B_i) = \left(\prod_{i \in I} A_i \right) \otimes \left(\prod_{i \in I} B_i \right).$$

Utilizing this we can infer that

$$\prod_{i=1}^{y+1} M_{2^{y+1}}^{(i)} = \left(\prod_{i=1}^{y} M_{2^{y+1}}^{(i)} \right) M_{2^{y+1}}^{y+1} = \left(\prod_{i=1}^{y} E_2 M_{2^y}^{(i)} \right) H_2 E_{2^y} =$$

$$= H_2 \otimes \left(\prod_{i=1}^{y} M_{2^y}^{(i)} \right) = H_2 \otimes H_{2^y} = H_{2^{y+1}}$$

thereby completing our proof.■

Now it is clear that we have a fast method to create H_{2^y}. It can be inferred that

$$H_{2^{y+1}} = H_2 \otimes H_{2^y} = \begin{pmatrix} H_{2^y} & H_{2^y} \\ H_{2^y} & -H_{2^y} \end{pmatrix}$$

and more generally

$$H_{2n} = H_2 \otimes H_n = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix}.$$

3.4.6 The decoding algorithm

Now we will examine the algorithm iself which will claim use of the method elaborated. Assume that the message is

$$u = (u_0, u_1 \dots u_{2^y-1})^T,$$

the encoded version is

$$v = G_{RM(1,y)}u = (v_0, v_1 \dots v_{2^y-1})^T,$$

and we receive

$$w = (w_0, w_2 \dots w_{2^y-1})^T.$$

Now our goal is do decode w. Let

$$W = ((-1)^{w_0}, (-1)^{w_1}, \dots (-1)^{w_{2^y-1}}).$$

Let c be the largest absolute value coordinate of WH_{2^y}. If $c = 2^m$, then $w = u$. If $c \neq 2^m$, then there is a series $(c_1, c_2, \dots c_y)$ that

$$|c| = \sum_{i=1}^{y} c_i 2^{i-1}.$$

Let $col_i(M)$ denote the i-th column of the matrix M. If $c > 0$, the codeword is

$$k = \sum_{i=1}^{y} c_i col_{y+2-i}(G_{RM(1,y)}),$$

Otherwise the codeword is $k + e$. Now we will see an example for this algorithm in the case of $RM(1,3)$. Say that the codeword

$$v^T = (1, 1, 0, 0, 0, 0, 1, 1)$$

is received as the word

$$w^T = (1, 0, 0, 0, 0, 0, 1, 1)$$

. It comes without difficulties that

$$W = (-1, 1, 1, 1, 1, 1, -1, -1).$$

Now we need to calculate the Hadamard matrix described in the abstract algorithm. It is obtained that

$$H_{2^3} = H_8 = H_2 \otimes H_4 = \begin{pmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{pmatrix} =$$

$$= \begin{pmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{pmatrix}$$

from where $WH_8 = (2, 2, -2, -2, 2, -6, -2, 2, -4)$, implying that $c = -6$ and $|c| = 6$. Since $-6 < 0$ and $c_1 = 0$, $c_2, c_3 = 1$ we conclude that the correct codeword is

$$e + 0 * (0, 1, 0, 1, 0, 1, 0, 1) + 1 * (0, 0, 1, 1, 0, 0, 1, 1) + 1 * (0, 0, 0, 0, 1, 1, 1, 1) =$$

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

This algortihm shows the importance of Hadamard matrices of higher orders. Now we will examine another method to create Hadamard matrices of higher orders. To do this we will introducte a concept similar to that of the Hadamard matrices.

3.4.7 Introduction to Conference matrices

Definition The $n \times n$ matrix C_n is by definition called a **conference matrix** of order n if and only if its diagonal only contains zeros, its other elements are either 1 or -1 and

$$C_n C_n^T = (n-1)E_n.$$

Obviously

$$C_1 = \begin{pmatrix} 0 \end{pmatrix}.$$

Let us find conference matrices of order 2. Let $a, b = \pm 1$. We have

$$\begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix} \begin{pmatrix} 0 & b \\ a & 0 \end{pmatrix} = \begin{pmatrix} a^2 & 0 \\ 0 & b^2 \end{pmatrix} = (2-1)E_2 = E_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

meaning that exatly the following are the Conference matrices of order 2

$$C_2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\widehat{C_2} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$\widehat{\widehat{C_2}} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

$$\widehat{\widehat{\widehat{C_2}}} = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}.$$

If C_n is a Conference matrix, then $-C_n$ is a conference matrix as well, since

$$(-C_n)(-C_n)^T = (-C_n)(-C_n^T) = C_n C_n^T = (n-1)E_n$$

directly from the definition. If C_n is a Conference matrix, then C_n^T is also a conference matrix, since

$$C_n C_n^T = (n-1)E_n \leftrightarrow C_n^T C_n = (n-1)E_n \leftrightarrow C_n^T (C_n^T)^T = (n-1)E_n.$$

The determinant of a Conference matrix is

$$\mathrm{Det}(C_n) = \pm (n-1)^{\frac{n}{2}}$$

because

$$C_n C_n^T = (n-1)E_n \rightarrow \mathrm{Det}(C_n C_n^T) = \mathrm{Det}((n-1)E_n) \leftrightarrow$$
$$\leftrightarrow \mathrm{Det}(C_n)\mathrm{Det}(C_n^T) = (n-1)^n \leftrightarrow \mathrm{Det}(C_n)\mathrm{Det}(C_n) = (n-1)^n \leftrightarrow$$
$$\leftrightarrow (\mathrm{Det}(C_n))^2 = (n-1)^n \leftrightarrow \mathrm{Det}(C_n) = \pm\sqrt{(n-1)^n} \leftrightarrow \mathrm{Det}(C_n) = \pm(n-1)^{\frac{n}{2}}.$$

It can be seen from the definition that every pair of different rows/columns are orthogonal. The inner product of two different rows are zero, meaning that the ones and minus ones cancell each other. Considering that in the main diagonal each row has exactly one zero, it means that $n-2$ elements are nonzero, meaning that $\frac{n-2}{2}$ ones and $\frac{n-2}{2}$ minus ones are present, consequently

$$\frac{n-2}{2} \in \mathbb{N} \leftrightarrow n-2 \equiv 0 \pmod 2 \leftrightarrow n \equiv 0 \pmod 2$$

which means that if C_n is a conference matrix, then n must be even. For the sake of completeness we provide the following two examples:

$$C_6 = \begin{pmatrix} 0 & +1 & +1 & +1 & +1 & +1 \\ +1 & 0 & -1 & +1 & +1 & -1 \\ +1 & -1 & 0 & -1 & +1 & +1 \\ +1 & +1 & -1 & 0 & -1 & +1 \\ +1 & +1 & +1 & -1 & 0 & -1 \\ +1 & -1 & +1 & +1 & -1 & 0 \end{pmatrix}$$

$$C_{10} = \begin{pmatrix} 0 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & 0 & -1 & -1 & -1 & +1 & +1 & -1 & +1 & +1 \\ +1 & -1 & 0 & -1 & +1 & -1 & +1 & +1 & -1 & +1 \\ +1 & -1 & -1 & 0 & +1 & +1 & -1 & +1 & +1 & -1 \\ +1 & -1 & +1 & +1 & 0 & -1 & -1 & -1 & +1 & +1 \\ +1 & +1 & -1 & +1 & -1 & 0 & -1 & +1 & -1 & +1 \\ +1 & +1 & +1 & -1 & -1 & -1 & 0 & +1 & +1 & -1 \\ +1 & -1 & +1 & +1 & -1 & +1 & +1 & 0 & -1 & -1 \\ +1 & +1 & -1 & +1 & +1 & -1 & +1 & -1 & 0 & -1 \\ +1 & +1 & +1 & -1 & +1 & +1 & -1 & -1 & -1 & 0 \end{pmatrix}$$

3.4.8 Constructing Hadamard matrices of higher orders with conference matrices

Theorem 3.7 *Let $C_{n,s}$ be a symmetric conference matrix of order n. It holds that*

$$H_{2n} = \begin{pmatrix} C_{n,s} + E_n & C_{n,s} - E_n \\ C_{n,s} - E_n & -C_{n,s} - E_n \end{pmatrix}$$

meaning that we have obtained a Hadamard matrix of order $2n$. Let $C_{n,as}$ be an antisymmetric conference matrix of order n. In this case we have

$$H_n = C_{n,as} + E_n.$$

Furthermore

$$H_{2n} = \begin{pmatrix} C_{n,as} + E_n & C_{n,as} + E_n \\ C_{n,as} + E_n & -C_{n,as} - E_n \end{pmatrix}.$$

Proof One can easily see that

$$C_{n,s}^2 = C_{n,s}C_{n,s} = C_{n,s}C_{n,s}^T = (n-1)E_n.$$

It is also clear that

$$(C_{n,s} + E_n)^2 + (C_{n,s} - E_n)^2 =$$
$$= C_{n,s}^2 + C_{n,s}E_n + E_nC_{n,s} + E_n^2 + C_{n,s}^2 - C_{n,s}E_n - E_nC_{n,s} + E_n^2 =$$
$$= 2C_{n,s}^2 + 2E_n^2 = 2(n-1)E_n + 2E_n = 2nE_n.$$

Using the aforementioned identities and the definition of conference matrices and Hadamard matrices we obtain that

$$\begin{pmatrix} C_{n,s} + E_n & C_{n,s} - E_n \\ C_{n,s} - E_n & -C_{n,s} - E_n \end{pmatrix} * \begin{pmatrix} C_{n,s} + E_n & C_{n,s} - E_n \\ C_{n,s} - E_n & -C_{n,s} - E_n \end{pmatrix}^T =$$

$$= \begin{pmatrix} C_{n,s} + E_n & C_{n,s} - E_n \\ C_{n,s} - E_n & -C_{n,s} - E_n \end{pmatrix} * \begin{pmatrix} C_{n,s} + E_n & C_{n,s} - E_n \\ C_{n,s} - E_n & -C_{n,s} - E_n \end{pmatrix} =$$

$$= \begin{pmatrix} (C_{n,s}+E_n)^2+(C_{n,s}-E_n)^2 & 0 \\ 0 & (C_{n,s}+E_n)^2+(C_{n,s}-E_n)^2 \end{pmatrix} =$$

$$= \begin{pmatrix} 2nE_n & 0 \\ 0 & 2nE_n \end{pmatrix} = 2n \begin{pmatrix} E_n & 0 \\ 0 & E_n \end{pmatrix} = 2nE_{2n}$$

holds which proves the first claim. For the second claim we can observe

$$-C_{n,as}^2 = -C_{n,as}C_{n,as} = C_{n,as}C_{n,as}^T = E_n$$

By easy calculation we infer that

$$(C_{n,as}+E_n)(C_{n,as}+E_n)^T = (C_{n,as}+E_n)(C_{n,as}^T+E_n^T) =$$

$$= (C_{n,as}+E_n)(-C_{n,as}+E_n) = -C_{n,as}^2 + C_{n,as}E_n - C_{n,as}E_n + E_n^2 =$$

$$-C_{n,as}^2 + E_n^2 = (n-1)E_n + E_n = nE_n.$$

The third claim can be proven utilizing the second claim and the identity $H_{2n} = H_2 \otimes H_n$. We have

$$H_{2n} = H_2 \otimes H_n = \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix} =$$

$$= \begin{pmatrix} C_{n,as}+E_n & C_{n,as} & C_{n,as}+E_n & C_{n,as} \\ C_{n,as}+E_n & C_{n,as} & -(C_{n,as}+E_n & C_{n,as}) \end{pmatrix} =$$

$$= \begin{pmatrix} C_{n,as}+E_n & C_{n,as}+E_n \\ C_{n,as}+E_n & -C_{n,as}-E_n \end{pmatrix}.$$

■

By the aforementioned theorem we have gained two formulas for $H_{2^{y+1}}$ as well. Firstly for symmetric matrices we have

$$H_{2^{y+1}} = \begin{pmatrix} C_{2^y,s}+E_{2^y} & C_{2^y,s}-E_{2^y} \\ C_{2^y,s}-E_{2^y} & -C_{2^y,s}-E_{2^y} \end{pmatrix}$$

and for antisymmetric matrices the formula is

$$H_{2^y} = \begin{pmatrix} C_{2^y,as}+E_{2^y} & C_{2^y,as}+E_{2^y} \\ C_{2^y,as}+E_{2^y} & -C_{2^y,as}-E_{2^y} \end{pmatrix}.$$

3.4.9 Conference matrices of higher order

Now we will briefly show how larger order conference matrices can be created by utilizing the theory of finite fields. Detailful examination is not our purpose now. Define $f : \mathbb{F}_q \to \{-1, 0, 1\}$ in the following way:

1. f(0)=0,
2. if $u \neq 0$ and u is a square of an element, then $f(u) = 1$,
3. for all other cases $f(u) = -1$.

Denote the elements of $\mathbb{F}_q$ by $z_0, z_1 \dots z_{q-1}$, where $z_0 = 0$. Thefine the matrix $Z(\mathbb{F}_q)$ in by the formula

$$z_{ij} = f(z_i - z_j)$$

where $0 \leq i, j \leq q-1$. Paley has shown in 1933 that

$$Z'(\mathbb{F}_q) = \begin{pmatrix} 0 & 1 & \cdots & 1 \\ \pm 1 & & & \\ \vdots & & Z(\mathbb{F}_q) & \\ \pm 1 & & & \end{pmatrix}$$

is a conference matrix of order $q+1$ where we chose the ± signs in a manner that if $q \equiv 1 \pmod 4$ then $Z'(\mathbb{F})$ is symmetric, and if $q \equiv -1 \equiv 3 \pmod 4$ then $Z'(\mathbb{F})$ then $Z(\mathbb{F})$ is antisymmetric. Now with this knowledge we can infer several things about the existence of certain Hadamard matrices. This is formally known as **Paley's Theorem**. If $q \equiv 1 \pmod 4$ then we have

$$H_{2(q+1)} = \begin{pmatrix} Z'(\mathbb{F}_q) + E_{q+1} & Z'(\mathbb{F}_q) - E_{q+1} \\ Z'(\mathbb{F}_q) - E_{q+1} & -Z'(\mathbb{F}_q) - E_{q+1} \end{pmatrix}$$

but if $q \equiv 3 \pmod 4$ we have

$$H_{q+1} = Z'(\mathbb{F}_q) + E_{q+1}.$$

3.5 Relationship with finite geometry

For this subsection I rely on [14] and [17]. Now we will examine that where the patterns of Reed-Muller codes appear in geometry.

Definition Let F be a field and V be a vector space over F. We can examine the geometric properties of the set

$$AG(V) = \{v + U \mid (v \in V) \wedge (U \leq V)\}.$$

This set is referred to as the **affine geometry of** V. Equivalently this is the set of all cosets over all subspaces of V. By definition

$$Dim(v + U) = Dim(U),$$

and we by definition we say that $v + U$ is a $Dim(v + U)$-flat of $AG(V)$. Let $I_{a,b}(AG(V))$ denote the incidence matrix of a-flats and b-flats of $AG(V)$.

The concept of **points** in Euclidean geometry are represented here by 0-flats, the **lines** are represented by 1-flats, and the **planes** by 2-flats. Now we will show some examples for this concept. The most trivial one would be $AG(\mathbb{F}_2)$. The definition clearly shows us that there are two points, namely $\{0\}$ and $\{1\}$. The only line will be $\{0, 1\}$. This means that we have two points and a line between them. How will $AG(\mathbb{F}_2^2)$ will look like? First let us find the points of this geometry. By definition we have to look for the $0 - flats$, and consequently we need to examine all the 0 dimensional subspaces. The only one is $\{00\}$, consequently our points are $\{00\}$, $\{01\}$, $\{10\}$, $\{11\}$. The one dimensional subspaces are $\{00, 01\}$, $\{00, 10\}$ and $\{00, 11\}$, therefore our lines will be $\{00, 01\}$, $\{10, 11\}$, $\{00, 10\}$, $\{01, 11\}$, $\{00, 11\}$, $\{01, 10\}$. The only two dimensional subset is $\{00, 01, 10, 11\}$, therefore the only plane will be $\{00, 01, 10, 11\}$. One can easily see that $AG(\mathbb{F}_2^2)$ can be described as a square, where points corresponds to the vertices of the square ,the lines correspond to the sides and the diagonals of the squares. The only plane contains all the points and lines. Considering this observation it becomes easier to give point-line the incidence matrix of $AG(\mathbb{F}_2^2)$. This incidence matrix is easy to calculate, because we know that each line contains exactly two points and for every pair of points there is exactly one line to contain them.

$$I_{0,2-1}(AG(\mathbb{F}_2^2)) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}.$$

We have already seen in 3, that the vectors of $I(AG(\mathbb{F}_2^2))$ all belong to $RM(1,2)$. If we take the span of those vectors, we obtain exactly the vectors of $RM(1,2)$. Let us see another example. It becomes clear that in the case of $AG(\mathbb{F}_2^3)$ the 8 points will correspond to the vertices of a cube, the lines will be the face diagonals, the space diagonals and the edges of the cube and the planes will be the faces of the cube and the planes containing the space diagonals. This perspective appears to be really practical when trying to figure out the point-plance incidence matrix of $AG(\mathbb{F}_2^3)$.

$$I_{0,3-1}(AG(\mathbb{F}_2^3)) = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Again, we have seen in 3, that these rows of $I_{0,3-1}(AG(\mathbb{F}_2^3))$ are all from $RM(1,3)$. The span of these vetors of course will give us the entire $RM(1,3)$.

Theorem 3.8 *The codewords of $RM(x,y)$ with minimum weight are exactly the rows of $I_{0,y-x}(AG(\mathbb{F}_2^y))$. Furthermode the rows of $I_{0,y-x}(AG(\mathbb{F}_2, y))$ span the entire $RM(x,y)$.*

3.6 Reed-Muller codes and multilinear polynomials

In this subsection the used source is [16].

Now we will look at an alternative view about Reed-Muller codes and thereby eleborating on the relationship with multilinear polynomials.

Definition Let $\mathbb{F}_q$ be a finite field and let I denote the ideal

$$(X_1^2 - X_1, X_2^2 - X_2, \dots X_x^2 - X_y)$$

in $\mathbb{F}_q[X_1, X_2, \dots X_y]$. The elements of the set

$$ML(\mathbb{F}_q, y) = \{f \bmod I \mid f \in \mathbb{F}_q[X_1, X_2, \dots X_y]\}$$

are called by definition **multilinear polynomials** over $\mathbb{F}_q$. In other words multilinear polynomials are multivariable polynomials whose every monomial is such that any X_i is at most at first exponent. Let

$$V : \mathbb{F}_2[X_1, X_2, \dots X_y] \to \mathbb{F}_2^{2^y}$$

be the function of substituting all elements of $\mathbb{F}_2^y$. Reed-Muller codes can be defined by

$$RM'(x,y) = \{V(f) \mid (f \in ML(\mathbb{F}_2, y)) \wedge (deg(f) \leq x)\}.$$

Because of the first isomorphism theorem we know that

$$Im(V) = \mathbb{F}_2^{2^y} \cong \mathbb{F}_2[X_1, X_2, \dots X_y]/I \cong ML(\mathbb{F}_2, y).$$

Theorem 3.9 *The two definition of the Reed-Muller codes are equivalent, meaning that they give the same codes. In other words for every (x, y) ordered pair we have*

$$RM(x, y) = RM'(x, y).$$

Proof We only need to prove that

$$RM'(x, y) \subseteq RM(x, y)$$

and that

$$dim(RM'(x, y)) = dim(RM(x, y)) = \sum_{i=0}^{x} \binom{y}{i}.$$

The statement concerning the dimension comes easily by using elementary combinatorics. For a basis of $RM'(x, y)$ can be constructed by monomials of degree s, where $0 \leq s \leq y$. The quantity of basis vector monomials with degree i is given by $\binom{y}{i}$, since we have y variables, and i of them needs to be selected. The first claim is equivalent with the statement that for all elements of $RM'(x, y)$ the recursion stated in the definition of $RM(x, y)$ holds. For $R(x, x)$ we have all the multilinear polynomials with x variables and with a degree at most x, therefore we obtain the entire $ML(\mathbb{F}_2, x)$. Substituting the elements of $\mathbb{F}_2^x$ into the polynomial we obtain $\mathbb{F}_2^{2^x}$ as stated in the recursion. The $(u, u + v)$ construction comes from the fact that for every $f \in ML(\mathbb{F}_2, x)$ polynomial there is a pair

$$(g, h) \in ML(\mathbb{F}_2, x - 1) \times ML(\mathbb{F}_2, x - 1)$$

such as

$$f(X_1, X_2, \dots X_y) = X_x g(X_1, X_2, \dots X_{y-1}) + h(X_1, X_2, \dots X_{y-1})$$

holds.If we substitute to f the word $(u, u + v)$ is obtained, where $u \in RM'(x, y - 1)$ and $v \in RM'(x - 1, y - 1)$ can be seen from the degree of the polynomials. ■

Now we will examine how a proof looks when starting from the polynomial definition and not from the recursive one. This might provide a different insight.

Theorem 3.10 *For all x and y we have*

$$d(RM'(x, y)) = 2^{x-y}.$$

Proof Now we will prove that $d(RM'(x, y)) = 2^{x-y}$. The weight of the codeword $V(f)$ by definition will be the number of points $P \in \mathbb{F}_2^x$ for which $f(P) \neq 0$. We can clearly see that

$$wt(RM'(x, y)) \leq wt\left(\prod_{i=1}^{y} X_i\right) = 2^{x-y}$$

since we are talking about those points whose first y coordinates are 1, and for the every other coordinates we have 2 choices independently. This means that

$$d(RM'(x, y)) = wt(RM'(x, y)) \leq 2^{x-y}.$$

Now we only need to show that every nonzero codeword has a weight of at least 2^{x-y}. Equivalently it is needed to be proven that for every nonzero element of

$$ML(\mathbb{F}_2, x)$$

with a degree not greater than y there is at least 2^{x-y} element in $\mathbb{F}_2^x$ such that the evaluation in those point is not zero. We will induct on x and y. If $y = 0$, then we are talking about nonzero constant polynomials, consequently all points of $\mathbb{F}_2^x$ is good for us, therefore we have at least $2^{x-y} = 2^{x-0} = 2^x$ points in which the polynomials are not evaluated zero. Assume that the theorem hold for polynomials with degree lesser than x or number of variables lesser than y. Let f be a multilinear polynomial with $x-1$ variables and $deg(f) = y$. Then there exists an

$$(g, h) \in ML(\mathbb{F}_2, y-1) \times ML(\mathbb{F}_2, y-1)$$

ordered pair of polynomials that

$$f(X_1, X_2, \dots X_y) = X_x g(X_1, X_2, \dots X_{y-1}) - h(X_1, X_2, \dots X_{y-1})$$

where

$$deg(g) \leq x-1 \wedge deg(h) \leq x$$

holds. The first case is when $h \equiv 0$. In this case $g \not\equiv 0$ since $f \not\equiv 0$. Then because of the induction hypothesis there are at least $2^{(x-1)-(y-1)} = 2^{x-y}$ points of $\mathbb{F}_2^{y-1}$ for which the evaluation in g does not give zero. utilizing the

$$(v_1, v_2, \dots v_{y-1}) \mapsto (v_1, v_2, \dots v_{y-1}, 1)$$

function, we have gained at least 2^{x-y} points in $\mathbb{F}_2^y$ which does not give zero evaluation in the case of f. The second case is when $g - h \equiv 0$. Because of $g \equiv h$ we have

$$f(X_1, X_2, \dots X_y) = (X_y - 1)g(X_1, X_2, \dots X_{y-1}).$$

Because of the induction hypothesis we have at least 2^{x-y} elements of $\mathbb{F}_2^{y-1}$ which does not give zero in g. Using

$$(v_1, v_2, \dots v_{y-1}) \mapsto (v_1, v_2, \dots v_{y-1}, 0)$$

we have obtained at least 2^{x-y} elements of $\mathbb{F}_2^y$ which does not give zero on f. The third case is when $h \not\equiv 0$ and $g - h \not\equiv 0$. For $X_m = 0$, then

$$f(X_1, X_2, \dots X_y) = -h(X_1, X_2, \dots X_{y-1}),$$

therefore there are at least 2^{x-1-y} appropriate point because of the induction hypothesis. For the $X_m = 1$ case

$$f(X_1, X_2, \dots X_x) = g(X_1, X_2, \dots X_{x-1}) - h(X_1, X_2, \dots X_{x-1})$$

holds, and because of the induction hypothesis, there are at least 2^{x-1-y} appropriate points. Since

$$2^{x-1-y} + 2^{x-1-y} = 2 * 2^{x-1-y} = 2^{x-y}$$

our proof is complete. ■

4 Generalized Reed-Muller Codes

4.1 Definition, examples, fundalemtal properties

To understand Generalized Reed-Muller codes I only claim to utilize [9].

Definition Let

$$RM_{gen}(q,x,y) \leq \mathbb{F}_q^{2^y}$$

denote the linear code defined by its generator matrix, which is the same as in the case of binary Reed-Muller codes, meaning that

$$G_{RM_{gen}(q,x,y)} = G_{RM(x,y)}.$$

It comes really easily from the definition that

$$RM_{gen}(2,x,y) = RM(x,y).$$

For every q

$$G_{RM_{gen}(q,1,1)} = G_{RM(1,1)} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

so to obtain our code we need to multiply this matrix with vectors of $\mathbb{F}_q^2$. Every element of $\mathbb{F}_q^2$ will be obtained, since the system

$$a \equiv a_1 \pmod{q} \wedge a + b \equiv a_2 \pmod{q}$$

always has a solution mod q. But the fact that

$$RM_{gen}(q,1,1) = \mathbb{F}_q^2$$

can be seen from

$$RM_{gen}(q,1,1) \leq \mathbb{F}_q^2$$

and

$$\mathrm{Card}(RM_{gen}(q,1,1)) = \mathrm{Card}(\mathbb{F}_q^2) = q^2.$$

Let us see how $RM_{gen}(3,1,3) \subseteq \mathbb{F}_3^{2^3} = \mathbb{F}_3^8$ looks like. We know that

$$G_{RM_{gen}(3,1,3)} = G_{RM(1,3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}.$$

It means that to obtain $RM_{gen}(3,1,3)$, we need to multiply $G_{RM_{gen}(3,1,3)}$ with the vectors of $\mathbb{F}_3^4$, therefore we obtain 81 codewords.

4.2 Complex numbers

For this subsection I used the following sources: [1], [2], [4]. Since the entire subsection 4.3.1 heavily relies on complex numbers, we will give a very brief summary of them.

4.2.1 Possible conceptualizations

Complex numbers can be comprehended and elaborated on a plethora of different ways. We will briefly summarize some of them.

- First, we can build the field of complex numbers from the field of real numbers. Let the set of real numbers be denoted by $\mathbb{R}$. The additive identity of the real numbers now will be denoted by $0_\mathbb{R}$ and the multiplicative identity by $1_\mathbb{R}$. All the operations over $\mathbb{R}$ will be indicated by the $\mathbb{R}$ symbol in the lower index. Consider the 7-tuple

  $$\mathbb{C} = (\mathbb{R}\times\mathbb{R}, (0_\mathbb{R}, 0_\mathbb{R}), (1_\mathbb{R}, 0_\mathbb{R}), +_\mathbb{C}, *_\mathbb{C}, \text{-}_\mathbb{C}, ^{-1_\mathbb{C}})$$

  with the operations

  $$+_\mathbb{C} : (\mathbb{R}\times\mathbb{R})\times(\mathbb{R}\times\mathbb{R}) \to (\mathbb{R}\times\mathbb{R})$$
  $$*_\mathbb{C} : (\mathbb{R}\times\mathbb{R})\times(\mathbb{R}\times\mathbb{R}) \to (\mathbb{R}\times\mathbb{R})$$
  $$\text{-}_\mathbb{C} : \mathbb{R}\times\mathbb{R} \to \mathbb{R}\times\mathbb{R}$$
  $$^{-1_\mathbb{C}} : (\mathbb{R}\times\mathbb{R})\setminus\{(0_\mathbb{R}, 0_\mathbb{R})\} \to \mathbb{R}\times\mathbb{R}$$

  defined in the following way:

  $$\forall((a_1,a_2),(a_3,a_4)) \in (\mathbb{R}\times\mathbb{R})\times(\mathbb{R}\times\mathbb{R}) : (a_1,a_2) +_\mathbb{C} (a_3,a_4) = (a_1 +_\mathbb{R} a_3, a_2 +_\mathbb{R} a_4)$$
  $$\forall((a_1,a_2),(a_3,a_4)) \in (\mathbb{R}\times\mathbb{R})\times(\mathbb{R}\times\mathbb{R}) :$$
  $$(a_1,a_2) *_\mathbb{C} (a_3,a_4) = (a_1 *_\mathbb{R} a_3 -_\mathbb{R} a_2 *_\mathbb{R} a_4, a_1 *_\mathbb{R} a_4 +_\mathbb{R} a_2 *_\mathbb{R} a_3)$$
  $$\forall(a_1,a_2) \in \mathbb{R}\times\mathbb{R} : \text{-}_\mathbb{C}(a_1,a_2) = (\text{-}_\mathbb{R}a_1, \text{-}_\mathbb{R}a_2)$$
  $$\forall(a_1,a_2) \in (\mathbb{R}\times\mathbb{R})\setminus\{(0_\mathbb{R},0_\mathbb{R})\} :$$
  $$(a_1,a_2)^{-1_\mathbb{C}} = (a/_\mathbb{R}(a *_\mathbb{R} a +_\mathbb{R} b *_\mathbb{R} b), \text{-}_\mathbb{R}(b/_\mathbb{R}(a *_\mathbb{R} a +_\mathbb{R} b *_\mathbb{R} b)))$$

  One can simply show that this 7-tuple is indeed a field. Define the functions

  $$\mathrm{Re} : (\mathbb{R}\times\mathbb{R}) \to \mathbb{R}$$
  $$\mathrm{Im} : (\mathbb{R}\times\mathbb{R}) \to \mathbb{R}$$

  in the following way:

  $$\forall(a_1,a_2) \in (\mathbb{R}\times\mathbb{R})\ \mathrm{Re}((a_1,a_2)) = a_1$$
  $$\forall(a_1,a_2) \in (\mathbb{R}\times\mathbb{R})\ \mathrm{Im}((a_1,a_2)) = a_2.$$

  The $\mathrm{Re}((a_1,a_2)) = a_1 \in \mathbb{R}$ number is called the **real part** of the complex number. The $\mathrm{Im}((a_1,a_2)) = a_1 \in \mathbb{R}$ number is called the **imaginary part** of the complex number.

  Obviously

  $$(a_1,a_2) = (\mathrm{Re}((a_1,a_2)), \mathrm{Im}((a_1,a_2))).$$

  The operation

  $$\mathrm{Conj} : \mathbb{R}\times\mathbb{R} \to \mathbb{R} \to \mathbb{R}$$

  defined by

  $$\forall(a_1,a_2) \in \mathbb{R}\times\mathbb{R} : \mathrm{Conj}((a_1,a_2)) = (a_1, \text{-}_\mathbb{R}a_2)$$

  is the **complex conjugate**. One can deduce without difficulty that

  $$|.|_\mathbb{C} : (\mathbb{R}\times\mathbb{R}) \to \mathbb{R}$$
  $$|(a_1,a_2)|_\mathbb{C} = \mathrm{Re}[(a_1,a_2) *_\mathbb{C} \mathrm{Conj}((a_1,a_2))]$$

  is an absolute value function over the field $\mathbb{C}$.

- We can easily understand the field of complex numbers as a field extension of the real field. If $\mathbb{R}$ denotes the field of real numbers, then
$$\mathbb{C} = \mathbb{R}(i)$$
In this case we can write the elements as
$$z = a + bi = \mathrm{Re}(z) + \mathrm{Im}(z)i \text{ where } a, b \in \mathbb{R} \wedge i^2 = -1.$$
- The field of complex numbers can also be obtained as the
$$\mathbb{R}[X]/(X^2 + 1)$$
quotient ring, which will behave as a field. In this case we create the modulo $X^2 - 1$ congruence classes and we build the arithmetic of $\mathbb{C}$ with those equivalence classes.
- The field of complex numbers can also be understood as a subring of $\mathbb{R}^{2\times 2}$ which behaves as a field. The matrices whose ring we need to observe are those contained by the set
$$H = \left\{ \begin{pmatrix} \Psi & -\widehat{\Psi} \\ \widehat{\Psi} & \Psi \end{pmatrix} \mid \Psi, \widehat{\Psi} \in \mathbb{R} \right\}.$$
Obviously the isomorphism is
$$z \mapsto \begin{pmatrix} \mathrm{Re}(z) & -\mathrm{Im}(z) \\ \mathrm{Im}(z) & \mathrm{Re}(z) \end{pmatrix} = \begin{pmatrix} \mathrm{Length}(z) \cdot \cos(\mathrm{Arg}(z)) & -\mathrm{Length}(z) \cdot \sin(\mathrm{Arg}(z)) \\ \mathrm{Length}(z) \cdot \sin(\mathrm{Arg}(z)) & \mathrm{Length}(z) \cdot \cos(\mathrm{Arg}(z)) \end{pmatrix}.$$

4.2.2 Polar form and exponential form

Let us be reminded that he complex number in polar form is given as
$$z = \mathrm{Length}(z) \cdot \Big[\cos(\mathrm{Arg}(z)) + i \cdot \sin(\mathrm{Arg}(z))\Big].$$
Furthermore let us also be reminded that the exponential form of a complex number is given by the formula
$$z = \mathrm{Length}(z) \cdot \exp(\mathrm{Arg}(z \cdot i)).$$

4.2.3 Roots of unity, primitive roots of unity

Now let $\mathbb{C}$ denote the set of complex numbers and the field of complex numbers at the same time.

Definition The $\zeta_n \in \mathbb{C}$ number is called an nth **complex root of unity** by definition if and only if ζ_n is a solution of the equation $x^n = 1$. The $\zeta_n \in \mathbb{C}$ number is called an nth **complex primitive root of unity** by definition if and only if ζ_n is a complex root of unity and the condition
$$\forall\ 1 \leq k < n : \zeta_n^k \neq 1$$
is satisfied.

Obviously the set
$$\mathrm{Z}_n = \left\{ \zeta_n = \cos\left(\frac{2k\pi}{n}\right) + i \cdot \sin\left(\frac{2k\pi}{n}\right) \mid k \in \mathbb{N}_{\leq n} \right\}$$
contains exactly the nth roots, and the set
$$\widehat{\mathrm{Z}}_n = \left\{ \zeta_n = \cos\left(\frac{2k\pi}{n}\right) + i \cdot \sin\left(\frac{2k\pi}{n}\right) \mid k \in \mathbb{N}_{\leq n} \wedge \mathrm{Gcd}(k, n) = 1 \right\}$$
contains exactly the nth primitive roots.

4.3 Connection with CCC.

The purpose of this subsection is to demonstrate an advanced theorem in connection with generalized Reed-Muller codes. On this subsection I was solely relying on [9].

4.3.1 Aperiodic autocorrelation and aperiodic cross-correlation function

Definition Let $a, b \in \mathbb{Z}_q^n$ be two sequences,

$$a = (a_0, a_1, \dots a_{n-1})$$

and

$$b = (b_0, b_1, \dots b_{n-1}).$$

Let us define the function

$$\rho_{cross} : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \times ([-n+1, n-1] \cap \mathbb{Z}) \to \mathbb{C}$$

by

$$\rho_{cross}(a, b, u) = \sum_{i=0}^{n-1-u} \zeta_q^{a_{i+u}-b_i}$$

if $0 \leq u \leq n-1$ and

$$\rho_{cross}(a, b, u) = \sum_{i=0}^{n-1+u} \zeta_q^{a_i-b_{i-u}}$$

otherwise, where

$$\zeta_q = \exp(2\pi i/q)$$

is a primitive complex root of unity with order q. This function is called an **aperiodic cross-correlation function**, and we say that u is the **displacement**. The

$$\rho_{auto}(a, u) = \rho_{cross}(a, a, u)$$

two variable function is by definition called an **aperiodic autocorrelation function**.

Examine this new concept with a really simple example. If $q = 2$, we have the only primitive complex root of unity

$$\zeta_2 = \exp(2\pi i/2) = \exp(\pi i) = -1.$$

Select $n = 2$. Now our function looks

$$\rho_{cross} : \mathbb{Z}_2^2 \times \mathbb{Z}_2^2 \times \{-1, 0, 1\} \to \mathbb{C}$$

defined by

$$\rho_{cross}(a, b, u) = \sum_{i=0}^{1-u} (-1)^{a_{i+u}-b_i}$$

if $u = 0$ or $u = 1$. and

$$\rho_{cross}(a, b, u) = \sum_{i=0}^{1+u} (-1)^{a_i-b_{i-u}}$$

for $u = -1$. Select $u = 1$, and let our two sequences be $(1, 0)$ and $(1, 1)$.

$$\rho_{cross}((1, 0), (1, 1), 1) = (-1)^0 = 1.$$

Now calculate the value for displacement $u = 0$.

$$\rho_{cross}((1,0),(1,1),0) = (-1)^0 + (-1)^1 = 0.$$

In the case of displacement $u = -1$

$$\rho_{cross}((1,0),(1,1),-1) = (-1)^0 = 1.$$

Roughly speaking the cross-correlation function is the degree of similarity between two series quantified. The variable u concerns the "lag" between the two series. If there is no lag, it is substituted that $u = 0$. So if there is no "lag", we have

$$\rho_{cross}(a,b,0) = \sum_{i=0}^{n-1} \zeta_q^{a_i - b_i}$$

so we are comparing the two series element by element with no displacement, and we are quanitfying the degree similarity. If the two series are utterly the same, we get

$$\rho_{cross}(a,a,0) = \rho_{auto}(a,0) = \sum_{i=0}^{n-1} \zeta_q^{a_i - a_i} = \sum_{i=0}^{n-1} \zeta_q^0 = \sum_{i=0}^{n-1} 1 = n.$$

Moreover because of the triangle inequality

$$\left|\rho_{cross}(a,b,u)\right| = \left| \sum_{i=0}^{n-1-u} \zeta_q^{a_{i+u}-b_i} \right| \leq \sum_{i=0}^{n-1-u} |\zeta_q^{a_{i+u}-b_i}| = \sum_{i=0}^{n-1-u} 1 = n - u$$

and

$$\left|\rho_{cross}(a,b,u)\right| = \left| \sum_{i=0}^{n-1+u} \zeta_q^{a_i-b_{i-u}} \right| \leq \sum_{i=0}^{n-1+u} |\zeta_q^{a_i-b_{i-u}}| = \sum_{i=0}^{n-1+u} 1 = n + u$$

and equality holds if and only if the two series are the same with the consideration of the displacement.

4.3.2 GCS and CCC

Definition Let $s(N,n) = \{s_0, s_1, \ldots s_{N-1}\}$ be a set of sequences, where every sequence is has n elements. $s(N,n)$ is by definition called a **GCS of order N** if and only if

$$\sum_{j=0}^{N} \rho(s_j, 0) = Nn$$

and for every $u \neq 0$ the equality

$$\sum_{j=0}^{N} \rho(s_j, u) = 0$$

holds. Let

$$S(N,N,n) = \{s_0(N,n), s_1(N,n), \ldots s_{N-1}(N,n)\} =$$
$$= \{\{s_0^0, s_1^0, \ldots s_{N-1}\}, \{s_0^1, s_1^1, \ldots s_{N-1}^1\}, \ldots \{s_0^{N-1}, s_1^{N-1}, \ldots s_{N-1}^{N-1}\}\}.$$

The set $S(N,N,n)$ is called a **CCC of order N** if and only if every set in the set is a GCS and for every distinct GCS the equality

$$\sum_{j=0}^{N-1} \rho(s_j^{j_1}, s_j^{j_2}, u) = 0$$

holds in the case of any pair of valid indices and displacement.

4.3.3 Construction of CCC codes via Generalized Reed-Muller codes

The following theorem will show a not elementary connection between CCC and the cosets of $RM_{gen}(q, x, y)$, namely how to construct CCCs from those aforementioned cosets.

Theorem 4.1 *We will utilize the notation*

$$\mathbb{N}_w = \{n \in N^+ \mid n \leq w\}.$$

Let $\mathbb{F}_q$ be a finitie field, and y be a positive integer. Let $k \leq y$ also be a positive integer, and the sets

$$I_1, I_2, \dots I_k$$

be a partition of $\mathbb{N}_y$. Let

$$\pi_\alpha : \mathbb{N}_{Card(I_\alpha)} \to I_\alpha$$

be a bijection for all valid α. Utilize the notation

$$A = (q/2) * \sum_{\alpha=1}^{k} \sum_{\beta=1}^{Card(I_\alpha)-1} v_{\pi_\alpha(\beta)} v_{\pi_\alpha(\beta+1)} +$$

$$+ \sum_{\alpha=2}^{k} \sum_{\beta=1}^{Card(I_\alpha)} \sum_{\tau=0}^{2^{\alpha-1}-1} \lambda_{\alpha,\beta,\tau} v_{\pi_\alpha(\beta)} \prod_{\gamma=1}^{\alpha-1} v_{\pi_\gamma(Card(I_\gamma))}^{\tau_\gamma}$$

where the other notation

$$v_i = (0, 0, 0, \dots 0, 1, 1, 1, \dots, 1, 0, 0, 0, \dots 0, 1, 1, 1 \dots 1, 0, 0, 0, \dots \dots 1)$$

where the consecutive sequences contain 2^{i-1} same characters and $1 \leq i \leq y$, and $\lambda_{\alpha,\beta,\tau} \in \mathbb{F}_q$. and the τ_i sequence denote the individual numbers in the binary representation of τ. Let

$$(n_1, n_2, \dots n_k)$$

be the binary representation of n and

$$(p_1, p_2, \dots p_k)$$

the binary representation of p. For all

$$c \in A + RM_{gen}(q, 1, y)$$

codeword with the notation

$$c_n^p = c + (q/2) * \left(\sum_{\alpha=1}^{k} n_\alpha v_{\pi_\alpha(1)} + \sum_{\alpha=1}^{k} p_\alpha v_{\pi_\alpha(Card(I_\alpha))} \right)$$

where

$$0 \leq n, k \leq 2^k - 1$$

and

$$G^p = \{c_0^p, c_1^p, \dots c_{2^k-1}^p\},$$

the set

$$\{G^0, G^1, \dots, G^{2^k-1}\}$$

forms a a CCC of order 2^k and length 2^y.

Proof We only have to check whether the set $G = \{G^0, G^1, \ldots, G^{2^k-1}\}$ satisfies the criteria for being a CCC. Since the definition has been uttered in a form of two criteria, we will represent our proof in two parts. In the first part of the proof we will show that for all valid index the set G^p is a GCS of order 2^k. In the second part of the proof we will prove that any pair of distinct sets satisfies the criteria concerning the cross-correlation function. So first let us check whether our sets are GCS. We only need to establish our case for $u \geq 0$, since

$$\sigma_{1,auto}(d,u) = \sigma_{2,auto} = (d,-u).$$

What we have to show is

$$\sum_{d \in G^p} \sigma_{auto}(d,u) = \sum_{d \in G^p} \sum_{i=0}^{2^y-1-u} \zeta_q^{d_{i+u}-d_i} = \sum_{i=0}^{2^y-1-u} \sum_{d \in G^p} \zeta_q^{d_{i+u}-d_i} = 0.$$

We will substitute $j = i + u$. Also we will refer to the binary representation of i as $(i_1, i_2, \ldots i_y)$ and the binary representation of j as $(j_1, j_2, \ldots j_y)$. If there exists an $0 \leq \alpha \leq k$ index for which $i_{\pi_\alpha(1)} \neq j_{\pi_\alpha(1)}$ holds, then for all $d \in G^p$ sequence there is a sequence $d' \in G^p$ which satisfies the conditions

$$d' = (d'_0, d'_1, \ldots d'_{2^y-1}) = d + q/2(v_{\pi_\alpha(2)})$$

and

$$d_j - d_i - d'_j + d'_i = q/2\big(i_{\pi_{\alpha(1)}} - j_{\pi_{\alpha(1)}}\big) =_q q/2.$$

At this point one can easily see that

$$\zeta_q^{d_j-d_i} / \zeta_q^{d'_j-d'_i} = \zeta_q^{d_j-d_i-(d'_j-d'_i)} = \zeta_q^{d_j-d_i-d'_j+d'_i} = \zeta_q^{q/2\left(i_{\pi_{\alpha(1)}}-j_{\pi_{\alpha(1)}}\right)} =$$

$$= \zeta_q^{q/2} = -1$$

from where it is obtained that

$$\zeta_q^{d_j-d_i} = -\zeta_q^{d'_j-d'_i}$$

which leads us to

$$\zeta_q^{d_j-d_i} + \zeta_q^{d'_j-d'_i} = 0$$

by which

$$\sum_{i=0}^{2^y-1-u} \sum_{d \in G^p} \zeta_q^{d_{i+u}-d_i} = \sum_{i=0}^{2^y-1-u} 0 = 0$$

can be inferred. Now let us examine the case when it is true that for all α indices $i_{\pi_{\alpha(1)}} = j_{\pi_{\alpha(1)}}$. Let

$$\widehat{\alpha} = Inf\Big\{\alpha \mid (\exists\beta)\big(i_{\pi_\alpha(\beta)} \neq j_{\pi_\alpha(\beta)}\big)\Big\}$$

and

$$\widehat{\beta} = Inf\Big\{\beta \mid i_{\pi_{\widehat{\alpha}}(\beta)} \neq j_{\pi_{\widehat{\alpha}}(\beta)}\Big\}.$$

Let i' be an integer which differs from i in the binary representation in only the position $\pi_{\widehat{\alpha}}(\widehat{\beta}-1)$ and let j' be an integer which differs from j in the position $\pi_{\widehat{\alpha}}(\widehat{\beta}-1)$. Now equivalently

$$i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} = 1 - i_{i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}}$$

and

$$j'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} = 1 - j_{i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}}.$$

Let $j' = i' + u$. By the definition of G we get that for all p and for all $d \in G^p$ there exists a sequence $(g_k)_{k=0}^{y}$ such that

$$d = A + \sum_{k=0}^{y} g_k v_k$$

holds. First let us look at the case when $\beta \geq 3$. From utilizing the definition of A formulated in the theorem and considering that i and i' differ in their binary representation only in the position $\pi_{\widehat{\alpha}}(\widehat{\beta} - 1)$ with a raw calculation we get to

$$d_i' - d_i = (q/2) * \Big(i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-2)} i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} - i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-2)} i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} + i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} i_{\pi_{\widehat{\alpha}}(\widehat{\beta})} -$$

$$- i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} i_{\pi_{\widehat{\alpha}}(\widehat{\beta})}\Big) + \sum_{\tau}^{2^{\widehat{\alpha}-1}} \lambda_{\widehat{\alpha},\widehat{\beta}-1,\tau} i'_{\pi_{\widehat{\alpha}}}(\widehat{\beta} - 1) \prod_{\gamma=1}^{\widehat{\alpha}-1} i^{\tau_\gamma}_{\pi_\gamma(Card(I_\gamma))} -$$

$$- \sum_{\tau}^{2^{\widehat{\alpha}-1}} \lambda_{\widehat{\alpha},\widehat{\beta}-1,\tau} i_{\pi_{\widehat{\alpha}}}(\widehat{\beta} - 1) \prod_{\gamma=1}^{\widehat{\alpha}-1} i^{\tau_\gamma}_{\pi_\gamma(Card(I_\gamma))} + g_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} i'_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} - g_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} =_q$$

$$=_q (q/2) * (i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-2)} + i_{\pi_{\widehat{\alpha}}(\widehat{\beta})}) + g_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}(1 - 2i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}) +$$

$$+ \sum_{\tau}^{2^{\widehat{\alpha}-1}} \lambda_{\widehat{\alpha},\widehat{\beta}-1,\tau}(1 - 2i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}) \prod_{\gamma=1}^{\widehat{\alpha}-1} i^{\tau_\gamma}_{\pi_\gamma(Card(I_\gamma))}$$

Now let us see the $\widehat{\beta} = 2$ case. The previous calculation can be adapded with the change that we will omit $\widehat{\beta} - 2$ from everywhere it appears. By the definition of $\widehat{\beta}$ we know that for any index smaller than $\widehat{\beta}$ the characters in the binary representation of i and j will be the same, meaning that

$$i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-2)} = j_{\pi_{\widehat{\alpha}}(\widehat{\beta}-2)}$$

and

$$i_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)} = j_{\pi_{\widehat{\alpha}}(\widehat{\beta}-1)}$$

furthermore because of the definition $\widehat{\alpha}$ we know that for smaller numbers the characters of i an j are the same for every β, meaning that

$$i_{\pi_\gamma(Card(I_\gamma))} = j_{\pi_\gamma(Card(I_\gamma))}$$

for all $\gamma < \widehat{\alpha}$ index. It can be seen now easily that most terms will be annihillated. We get

$$d_j - d_i - d_j' - d_i' = (q/2) * (i_{\pi_{\widehat{\alpha}}(\widehat{\beta})} - j_{\pi_{\widehat{\alpha}}(\widehat{\beta})}) =_q q/2$$

and because of the reasons seen in the first case it can be concluded that every G^p is a GCS. Now we have arrived to the second part of the proof. Let us utilize the notation

$$c_n^p = (c_{n,0}^p, c_{n,1}^p, \ldots c_{n,2^y-1}^p).$$

What is needed to be shown is that for every GCS the equality

$$\sum_{n=0}^{2^k-1} \rho_{cross}(c_n^{p_1}, c_n^{p_2}, u) = \sum_{n=0}^{2^k-1} \sum_{i=0}^{2^y-1-u} \zeta_q^{c^{p_1}_{n,i+u} - c^{p_2}_{n,i}} = \sum_{i=0}^{2^y-1-u} \sum_{n=0}^{2^k-1} \zeta_q^{c^{p_1}_{n,i+u} - c^{p_2}_{n,i}} = 0$$

holds for every positive displacement. Now let us first examine the case when there exists anα number such that

$$i_{\pi_\alpha(1)} \neq j_{\pi_\alpha(1)}$$

holds. Then for $p \in \{p_1, p_2\}$ there exist sequences c'^{p}_{n} such that both

$$c^{p}_{n'} = c^{p}_{n} + (q/2)v_{\pi_\alpha(1)} \in G^p$$

and

$$c^{p_1}_{n,j} - c^{p_2}_{n,i} - c^{p_1}_{n',j} + c^{p_2}_{n',i} = (q/2) * (i_{\pi_\alpha(1)} - j_{\pi_\alpha(1)}) =_q q/2$$

hold. Now similarly to the first case of the first part we arrive to our conclusion. Let us examine the case when

$$i_{\pi_\alpha(1)} = j_{\pi_\alpha(1)}$$

hold for all indices. We will utilize the same notations as in the second part of the first case. It comes from a raw calculation similar to that in the second case of the first part that

$$c^{p_1}_{n,j} - c^{p_2}_{n,i} - c^{p_1}_{n,j'} + c^{p_2}_{n,i'} =_q q/2.$$

It follows then that

$$\zeta_q^{c^{p_1}_{n,j} - c^{p_2}_{n,i}} / \zeta_q^{c^{p_1}_{n,j'} - c^{p_2}_{n,i'}} = \zeta_q^{c^{p_1}_{n,j} - c^{p_2}_{n,i} - (c^{p_1}_{n,j'} - c^{p_2}_{n,i'})} =$$
$$\zeta_q^{c^{p_1}_{n,j} - c^{p_2}_{n,i} - c^{p_1}_{n,j'} + c^{p_2}_{n,i'}} = \zeta_q^{q/2} = -1$$

implying that

$$\zeta_q^{c^{p_1}_{n,j} - c^{p_2}_{n,i}} = -\zeta_q^{c^{p_1}_{n,j'} - c^{p_2}_{n,i'}}$$

meaning that

$$\sum_{n=0}^{2^k-1} \rho_{cross}(c^{p_1}_{n}, c^{p_2}_{n}, u) = \sum_{i=0}^{2^y-1} \sum_{n=0}^{2^k-1} (\zeta_q^{c^{p_1}_{n,j} - c^{p_2}_{n,i}} + \zeta_q^{c^{p_1}_{n,j'} - c^{p_2}_{n,i'}}) = 0$$

The $u < 0$ case comes similarly. We now only need to establish our case for zero displacement, in formula

$$\sum_{n=0}^{2^k-1} \rho_{cross}(c^{p_1}_{n}, c^{p_2}_{n}, 0) = \sum_{n=0}^{2^k-1} \sum_{i=0}^{2^y-1} \zeta_q^{c^{p_1}_{n,i} - c^{p_2}_{n,i}}.$$

Let

$$(p_{1,1}, p_{1,2}, \dots p_{1,k})$$

be the binary representation of p_1 and

$$(p_{2,1}, p_{2,2}, \dots p_{2,k})$$

be the binary representation of p_2. For all $n \le 2^k$ it holds that

$$c^{p_1}_{n} - c^{p_2}_{n} =_p (q/2) * \left(\left(\sum_{j=1}^{k} (p_{1,k} + p_{2,k}) v_{\pi_k(Card(I_k))} \right) \bmod 2 \right)$$

and $wt(d) = 2^{y-1}$, consequently it comes easily that

$$Card\Big\{ (c^{p_1}_{n,i}, c^{p_2}_{n,i}) \mid \zeta_q^{c^{p_1}_{n,i} - c^{p_2}_{n,i}} = \zeta_q^{q/2} = -1 \wedge (0 \le i \le 2^y - 1) \Big\} = 2^{y-1}$$

and

$$Card\Big\{ (c^{p_1}_{n,i}, c^{p_2}_{n,i}) \mid \zeta_q^{c^{p_1}_{n,i} - c^{p_2}_{n,i}} = \zeta_q^{0} = 1 \wedge (0 \le i \le 2^y - 1) \Big\} = 2^{y-1}$$

consequently the elements in the sum

$$\sum_{n=0}^{2^k-1} \rho_{cross}(c^{p_1}_{n}, c^{p_2}_{n}, 0)$$

are annihillating each other, completing our proof. ■

5 Other Codes

This entire section is dedicated to give a very brief inshight to other types of codes.

5.1 Reed-Solomon Codes

For this subsection I utilized [16].

5.1.1 Definition, examples, encoding

The Reed-Solomon Codes are a specific case of the Reed-Muller codes, although because of their conspicuous historical and mathematical relevance they need to be mentioned as a separate one. The generalization occurs by extending the code to multivariable polynomials.

Definition Let $n \in \mathbb{N}$ and $k \in [1, n[\cap \mathbb{N}$. Let $\mathbb{F}$ be a field with $\mathrm{Card}(F) \geq n$ and

$$S = \{a_i \mid 1 \leq i \leq n\} \subseteq \mathbb{F}.$$

The **Reed-Solomon code** is defined in the following way:

$$RS_{\mathbb{F},S}[n,k] = \{(p(a_1), \dots, p(a_n)) \in \mathbb{F}^n \mid p \in \mathbb{F}[X] \wedge \deg(p) \leq k-1\}$$

The Reed-Solomon codes are linear codes over $\mathbb{F}$. Let the original message to be

$$m = (m_0, m_1, \dots m_{k-1}) \in \mathbb{F}^k.$$

The polynomial corresponding to the message is

$$p_{(m)}(X) = \sum_{\gamma=0}^{k-1} m_\gamma X^\gamma = m_0 + m_1 X + \dots + m_{k-1} X^{k-1} \in \mathbb{F}[X].$$

Then for all i we calculate

$$p_{(m)}(a_i) = \sum_{\gamma=0}^{k-1} m_\gamma a_i^\gamma = m_0 + m_1 a_i + \dots + m_{k-1} a_i^{k-1} \in \mathbb{F}[X]$$

consequently the encoded version has the form

$$\left(\sum_{\gamma=0}^{k-1} m_\gamma a_i^\gamma \right)_{i=1}^{n} = \left(\sum_{\gamma=0}^{k-1} m_\gamma a_1^\gamma, \sum_{\gamma=0}^{k-1} m_\gamma a_2^\gamma, \cdots \sum_{\gamma=0}^{k-1} m_\gamma a_n^\gamma \right).$$

Evaluating our polynomial on points $a_1, a_2, \dots a_n$ is equivalent to multiplying with

$$G_{RS_{\mathbb{F},S}[n,k]} = \begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_n & a_n^2 & \dots & a_n^{k-1} \end{pmatrix}$$

meaning that the message m will be encoded as

$$\begin{pmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_n & a_n^2 & \dots & a_n^{k-1} \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \end{pmatrix} = \begin{pmatrix} \sum_{\gamma=0}^{k-1} m_\gamma a_1^\gamma \\ \sum_{\gamma=0}^{k-1} m_\gamma a_2^\gamma \\ \vdots \\ \sum_{\gamma=0}^{k-1} m_\gamma a_n^\gamma \end{pmatrix}.$$

Now we will briefly examine some elementary properties of Reed-Solomon codes.

5.1.2 Distance of Reed-Solomon codes

Theorem 5.1

$$d(RS_{\mathbb{F},S}[n,k]) = n - k + 1$$

Proof We only have to look at how many zeros can a codeword have at maximum. This is obviously equivalent to the question that how many roots can an univariable polynomial over $\mathbb{F}$ have. Our polynomials are non-zero and have a degree at most $k-1$. It immediately implies that such polynomial can have at most $k-1$ roots over $\mathbb{F}$, meaning that our codeword has at most $k-1$ zero coordinates. Since $n-(k-1) = n-k+1$, it can be inferred that our codeword has at least $n-k+1$ nonzero elements, from which one can easily see that the codeword containing the most zeros has at least $n-k+1$ nonzero elements, meaning that

$$d(RS_{\mathbb{F},S}[n,k]) \geq n - k + 1.$$

The equality part can be obtained directly from the singleton bound. ■

Theorem 5.2

$$Dim(RS_{\mathbb{F},S}[n,k]) = k$$

Proof This one easily follows from the fact that $\text{Rank}(G_{RS_{\mathbb{F},S}[n,k]}) = k$. ■

It can be shown as well that

$$RS_{\mathbb{F},S}[n,k] = \left\{ (c_0, c_1, \dots c_{n-1}) \in \mathbb{F}^n \mid c(X) = \sum_{\gamma=1}^{n-1} c_\gamma X^\gamma \wedge \forall\ 1 \leq \gamma \leq n-k : c(a^\gamma) = 0 \right\}$$

where $a \in \mathbb{F} \setminus \{0\}$ is a primitive element

$$S = \{1, a, \dots a^{n-1}\}$$

furthermore

$$\text{Card}(\mathbb{F}) = n + 1.$$

It means that the polynomials which correspond to the codewords of a Reed-Solomon code are exactly those which vanish at the

$$1, a, a^2 \dots a^{n-k}$$

powers of the primitive element a.

5.1.3 Common applications

Reed-Solomon codes were and are utilized in a plethora of different scenarios. Some of them will be listed below.

- Storage devices such as CDs, DVDs, HDDs still use them.
- They are still utilized in space transmission.
- Bar codes use them as well.
- It has indirect applications stemming from its generalized versions, for instance the Reed-Muller codes.

5.2 BCH Codes

For this subsection I relied on solely [16]. The BCH Codes are indeed a generalization of the Reed-Solomon Codes.

Definition Let

$$n = 2^m - 1$$

and d be the distance of the code, furthermore let

$$a \in \mathbb{F}_{2^m} \setminus \{0\}$$

be a primitive element. Te **binary BCH Code** can be defined as

$$\mathrm{BCH}[n, d] = \left\{ (c_0, c_1, \ldots c_{n-1}) \in \mathbb{F}_2^n \mid c(X) = \sum_{\gamma=0}^{n-1} c_\gamma X^\gamma \wedge\ 1 \leq \forall\ \gamma \leq d-1 : c(a^\gamma) = 0 \right\}$$

The BCH codes have a plethora of really interesting properties. For instance it holds that

$$\mathrm{BCH}[n, d] = \mathrm{RS}_{\mathbb{F}, \mathbb{F} \setminus \{0\}}[n, n-d+1] \cap \mathbb{F}_2^n$$

Furthermore it can be shown that

$$\mathrm{Dim}(\mathrm{BCH}[n, d]) \geq n - \left\lfloor \frac{d-1}{2} \right\rfloor \log(n+1)$$

BCH codes have a plethora of different applications just as Reed-Solomon codes.

5.3 Hamming Codes

I used only [20] for this subsection.

Definition Let $\mathbb{F}_q^\Psi$ a finite vector space over the finite field $\mathbb{F}_q$. Let

$$\mathrm{vs} = v_1, v_2, \ldots v_n$$

be a sequence of vectors such that all of these vectors form an one-dimensional subspace. Then the **Hamming codes** are defined as

$$\mathrm{HAMMING}(\mathbb{F}_q, \Psi, \mathrm{vs}) = \left\{ (\lambda_1, \lambda_2, \ldots \lambda_n) \in \mathbb{F}_q^n \mid \sum_{\gamma=0}^{n} \lambda_\gamma v_\gamma = 0 \right\}$$

Indeed, it comes easily that

$$n = \frac{q^\Psi - 1}{q - 1} = \sum_{\gamma=1}^{\Psi-1} q^\gamma = 1 + q + \cdots + q^{\Psi-1}$$

since

$$\mathrm{Card}(\mathbb{F}_q^\Psi \setminus \{0\}) = q^\Psi - 1$$

and we have

$$\mathrm{Card}(\mathbb{F}_q \setminus \{0\}) = q - 1.$$

We can infer without difficulties that

$$n = \frac{2^{\Psi} - 1}{2 - 1} = 2^{\Psi} - 1 = \sum_{\gamma=1}^{\Psi-1} 2^{\gamma} = 1 + 2 + \cdots + 2^{\Psi-1}$$

holds for binary Hamming Codes. Let us consider an elementary example. Our fiinite field will be

$$\mathbb{F}_4 = \{0, 1, A, B\}$$

and we will have $\Psi = 2$. Since $q = 4$ we have

$$n = \frac{4^2 - 1}{4 - 1} = \frac{15}{3} = 5$$

meaning that we will need 5 vectors. Obviously we have

$$v_1 = (0, 1)$$

$$v_2 = (1, 0)$$

$$v_3 = (1, 1)$$

$$v_4 = (1, A)$$

$$v_5 = (1, B)$$

and

$$\mathrm{vs} = ((0, 1), (1, 0), (1, 1), (1, A), (1, B)).$$

Since

$$(0, 1) + (1, A) + (1, B) = (1 + 1, A + B + 1) = (0, 1 + 1) = (0, 0)$$

$$(1, 0) + B(1, A) + A(1, B) = (1, 0) + (B, 1) + (A, 1) = (1 + B + A, 1 + 1) = (1 + 1, 0) = (0, 0)$$

$$(1, 1)+A(1, A)+B(1, B) = (1, 1)+(A, B)+(B, A) = (1+A+B, 1+B+A) = (1+1, 1+1) = (0, 0).$$

Additionally we can easily figure out that the codewords now obtained from these linear combinations span the entire code. It means that our code is three dimensional and a basis is

$$(1, 0, 0, 1, 1)$$

$$(0, 1, 0, B, A)$$

$$(0, 0, 1, A, B).$$

$$\mathrm{HAMMING}(\mathbb{F}_4, 2, ((0, 1), (1, 0), (1, 1), (1, A), (1, B))) =< (1, 0, 0, 1, 1), (0, 1, 0, B, A), (0, 0, 1, A, B) >$$

Now we will look at how the simpler Hamming Codes look like. Choose our field to be $\mathbb{F}_2$ and $\Psi = 2$. In this case we have

$$n = \frac{2^2 - 1}{2 - 1} = 3$$

vectors. These vectors are indeed

$$v_1 = (0, 1)$$

$$v_2 = (1, 0)$$

$$v_3 = (1, 1)$$

And the only nonzero codeword is

$$(1, 1, 1)$$

since

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

is the only nontrivial linear combination which gives back the zero vector. Now not only we have found the basis of the Hamming code, we found all nonzero codes as well. Obviously

$$\mathrm{RC}(3) = \mathrm{HAMMING}(\mathbb{F}_2, 2, ((0,1),(1,0),(1,1))) = \{(0,0,0),(1,1,1)\}$$

obtaining the repetition code of length 3. Let us observe a more trivial Hamming code. Now our field will be $\mathbb{F}_2$ again, but $\Psi = 1$. It means that

$$n = \frac{2^1 - 1}{2-1} = 1$$

which means that we will have only one vector, namely $v_1 = (1)$. It means that we will have no nonzero codewords, therefore every message will be decoded as 0. Indeed, one can easily see that the dimension of the Hamming code is

$$\mathrm{Dim}(\mathrm{HAMMING}(\mathbb{F}_q, \Psi, \mathrm{vs})) = n - \Psi = \frac{q^\Psi - 1}{q-1} - \Psi.$$

Indeed we can see as well that $d = 3$.

5.3.1 Hamming codes are perfect codes

Theorem 5.3 *The Hamming codes are perfect codes.*

Proof Since the dimension of the Hamming code is $n - \Psi$ and it is a vector space over $\mathbb{F}_q$, we have $q^{n-\Psi}$ codewords. The disjoint spheres will be exactly those whose radius is 1. We can alter exactly one coordinate to remain within the sphere. Altering two coordinates would get us to the 1-radius sphere of another codeword, since altering three coordinates would result in another codeword. So if we want to remain the 1 radius sphere of the codeword, we can only modify 1 coordinate. Since every codeword has n letters, we have n options. For every position the letter can be q different letter, consequently we have $q-1$ ways to alter it. Altogether it means that we can modify the codeword $n(q-1)$ different ways to remain within the 1 radius sphere. We can do it for all the codewords, resulting in

$$n(q-1)q^{n-\Psi}$$

vectors to be covered aside from the centers. If we calculate the centers as well, we have

$$n(q-1)q^{n-\Psi} + q^{n-\Psi}$$

points that the disjoint circles cover. Simplifying the expression we obtain

$$n(q-1)q^{n-\Psi} + q^{n-\Psi} = \frac{q^\Psi - 1}{q-1}(q-1)q^{n-\Psi} + q^{n-\Psi} = (q^\Psi - 1)q^{n-\Psi} + q^{n-\Psi} =$$

$$= q^\Psi q^{n-\Psi} - q^{n-\Psi} + q^{n-\Psi} = q^\Psi q^{n-\Psi} = q^{\Psi+n-\Psi} = q^n$$

meaning that all q^n vectors of the vector space is covered by the disjoint spheres. By definition it means that the Hamming codes are perfect codes. ■

References

[1] Kiss Emil *Bezetetés az Algebrába.* Typotex, Budapest, Hungary, 2007. https://people.inf.elte.hu/nebsabi/2011-2012-1/Algebra/Kiss%20Emil%20-%20Algebra.pdf

[2] Kristóf János *A Matematikai analízis alapjai* http://web.cs.elte.hu/~krja/analyse/a0.pdf

[3] Kristóf János *A Matematikai analízis elemei I.* http://web.cs.elte.hu/~krja/analyse/a1.pdf

[4] Kiss Emil Freely Available Notes for the course Algebra1 http://ewkiss.web.elte.hu/wp/wordpress/oktatas/faliujsag/a-regebbi-felevek-anyagai/eloadas-algebra1-normal-2017-osz/

[5] Kiss Emil Freely Available Notes for the course Algebra2 http://ewkiss.web.elte.hu/wp/wordpress/oktatas/faliujsag/a-regebbi-felevek-anyagai/eloadas-algebra2-normal-2018-tavasz/

[6] Professor G. David Forney *EE392D - Channel Coding: Techniques, Analysis and Design Principles - Winter 2007* https://web.stanford.edu/class/ee392d/Chap7.pdf

[7] Institute of Communications Engineering, National Sun Yat-sen University, Notes Belongling to Wireless Information Transmission System Lab. *Chapter 5: Cyclic Codes* http://apwcs2014.nsysu.edu.tw/course/pdfdownload/95_2/%E9%8C%AF%E8%AA%A4%E6%9B%B4%E6%AD%A3%E7%A2%BC/CC-04-CyclicCode.pdf

[8] Sarah A. Spence *Introduction to Algebraic Coding Theory Supplementary material for Math 336 Cornell University.* https://pdfs.semanticscholar.org/2581/928b53ea8f374d4a32d1b1ba53814cc9d29b.pdf Cornell University, 2002, Freely available PDF teaching/studying material

[9] Chao-Yu Chen, Chung-Hsuan Wang, and Chi-chao Chao *Complete Complementary Codes and Generalized Reed-Muller Codesl.* https://ir.nctu.edu.tw/bitstream/11536/8230/1/000260956700017.pdf IEEE COMMUNICATIONS LETTERS, VOL. 12, NO. 11, NOVEMBER 2008

[10] SIDDHARTHA BISWAS *INTRODUCTION TO CODING THEORY: BASIC CODES AND SHANNON'S THEOREM.* http://www.math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Biswas.pdf

[11] Henk C.A. van Tilborg *Coding Theory : A First Course.* 1993, Freely available PDF teaching/studying material http://hyperelliptic.org/tanja/teaching/CCI11/CODING.pdf

[12] Yehuda Lindell *Introduction to Coding Theory Lecture Notes.* Department of Computer Science Bar-Ilan University, Israel, January 25, 2010 http://u.cs.biu.ac.il/~lindell/89-662/coding_theory-lecture-notes.pdf

[13] SAN LING and CHAOPING XING *Coding Theory A First Course.* http://site.iugaza.edu.ps/mashker/files/coding-theory-a-first-course.pdf

https://www.cambridge.org/hu/academic/subjects/mathematics/discrete-mathematics-information-theory-and-coding/coding-theory-first-course?format=HB&isbn=9780521821919 Cambridge University Press 2004, National University of Singapore

[14] Sebastian Raaphorst Carleton University *Reed-Muller Codes* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.3214&rep=rep1&type=pdf May 9, 2003

[15] YANG WANG *Hadamard Matrices and Reed-Muller Codes* https://www.math.ust.hk/~yangwang/Course/2016FSMath4999/Kin%20Li/Capstone2016.pdf. Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, +852 2358-7412 (W), yangwang@ust.hk

[16] Eric Blais and Venkat Guruswami *Introduction to Coding Theory CMU: Spring 2010.* https://www.cs.cmu.edu/~venkatg/teaching/codingtheory/notes/notes6.pdf Notes 6: Reed-Solomon, BCH, Reed-Muller, and concatenated codes February 2010

[17] Bill Cherowitzo *Reed-Muller Codes* http://www-math.ucdenver.edu/~wcherowi/courses/m7823/reedmuller.pdf

[18] Massoud Malek *Reed-Muller Codes* http://www.mcs.csueastbay.edu/~malek/Class/Reed-Muller.pdf California State University, East Bay

[19] Hadamard and conference matrices Peter J. Cameron *Hadamard and conference matrices* University of St Andrews and Queen Mary University of London http://www.maths.qmul.ac.uk/~whitty/LSBU/MathsStudyGroup/cameron-oct14.pdf

[20] Robert A. Wilson *The Golay code* 01/12/08, QMUL, Pure Mathematics Seminar http://www.maths.qmul.ac.uk/~raw/talks_files/Golay.pdf