MA 576
Introduction to Codes and their use

Jürgen Bierbrauer

February 8, 1999

Contents

Chapter 1

Introduction

Historically Coding Theory originated as the mathematical foundation for the transmission of messages over noisy channels. While this type of application is still very important and popular the last decades have shown that the applications of codes are not restricted to this setting. In fact, a multitude of diverse applications have been discovered. In this text we want to give a modern introduction to the mathematical theory of codes and to some of its applications.
We start with a description of the setting, which led to the development of coding theory: a **transmitter** T wishes to send a **message** to the **receiver** R. What is this message? We imagine there is only a finite number of possible messages and that T and R know beforehand the set of all possible messages. If for example there are 16 possible messages, then each message can be described as a binary quadrupel. Each message has then the form 0000 or 0110 or 1011 or ... We say four **bits** have to be sent. Messages may of course be very much longer. Assume we want to send a black and white picture. Divide it into 10,000 pixels, where each pixel has one of $256 = 2^8$ shades (for example, 00000000 for white, 11111111 for black ...). In that case we would need to send 80,000 bits in order to transmit the picture. Where is the problem? We imagine T and R to be seperated in space or time, so that transmission errors may occur. For example, if 0110 is sent and an error occurs in the third coordinate, then 0100 will be received. Moreover we treat these errors as being randomly generated. The probabilistic rules according to which errors occur determine the nature of the **channel.** A popular and particularly simple model is the **binary symmetric channel BSC:** The

channel is called binary as it transmits only one of two symbols (0 or 1) at a time. If a 0 is sent, then a 1 will be received with probability p. If a 1 is sent, then 0 will received with the same probability p. The probability p is fixed. It is clear that only cases $0 \leq p < 1/2$ make sense. This channel is called symmetric, as the probability of an error is independent of the symbol sent. One also speaks of a **memoryless** channel, for obvious reasons.
The objective of coding theory in this setting is to provide means for the reliable transmission of messages. Consider the case of 16 possible messages to be sent via a BSC. The probability that a message transmits correctly is clearly $(1-p)^4$. Here is an easy method to improve upon this, which just about everybody would think of eventually, the **repetition code**: repeat each symbol three times. Thus, if the message is 1101, we would transmit 111111000111. The receiver will decode each block of three symbols by a majority decision, for example

$$000 \mapsto 0, 111 \mapsto 1, 110 \mapsto 1, 001 \mapsto 0.$$

This is based on the observation that for each symbol an error is less probable than a non-error (corresponding to $p < 1/2$). A symbol 0 in the original message will be decoded as a 0 if and only if at most one of the symbols in the corresponding block 000 transmits incorrectly. In that sense our repetition code will correct one error. The probability for this is $(1-p)^3 + 3p(1-p)^2 = (1-p)^2(1+2p)$. The probability that a message is decoded correctly is therefore $(1-p)^8(1+2p)^4$. If this probability should be too high we could use blocks of length $5, 7, \ldots$ The drawback of this method is, that the message to be sent via the channel becomes very long.
By the way, we have two conflicting meanings for the word **message** now. In order to remedy this it is customary to reserve the word message for the signals sent through the channel. The signals, which the transmitter wishes to send to the receiver, are called **source states.** In the case of the repetition code the source state was 1101, the message 111111000111.
As long messages cost time and money we are in the situation of a trade-off: we want to transmit reliably, but we also wish to limit the length of the messages. It is clear that the repetition code is not very sophisticated. Here is our first basic definition:

Definition (1.1) (code) *Let $\mathcal{A}$ be a finite set of v elements (the* **alphabet***). A v-ary code $\mathcal{C}$ of* **length** *n is a family of n-tuples with entries in $\mathcal{A}$:*

$$\mathcal{C} \subseteq \mathcal{A}^n.$$

Definition (1.2) (distance) *Let $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n) \in \mathcal{A}^n$. The* **(Hamming) distance** *between x and y is*

$$d(x, y) = \text{ number of } i \text{ such that } x_i \neq y_i.$$

The **minimum distance** *$d(\mathcal{C})$ of code $\mathcal{C} \subseteq (\mathcal{A})^n$ is the minimum of the Hamming distances $d(x, y)$, where $x, y \in \mathcal{C}, x \neq y$.*

It is easy to see that the Hamming distance is a metric. Its meaning in our setting is obvious: if $x \in \mathcal{A}^n$ is sent and $y \in \mathcal{A}^n$ is received, then at least $d(x, y)$ errors must have occured, and the most probable explanation is that precisely $d(x, y)$ errors have occured. Here is how codes are used for information transmission: T and R agree on a code $\mathcal{C} \subseteq \mathcal{A}^n$ of minimum distance d. The tuples of $\mathcal{C}$ are called **code words.** Naturally there must be at least as many code words as source states. Consider the case $d = 3$. Let $y = (y_1, y_2, \ldots, y_n)$ be the received vector. Observe that R knows a code word x was sent. If $y \in \mathcal{C}$, then R will decode $y \mapsto x = y$. Moreover this will be correct unless at least 3 errors have occured. Assume $y \notin \mathcal{C}$. Then R will search for a code word x at distance 1 from y. There can be at most one such word (if x' was a second such code word, then $d(x, x') \leq 2$, contradiction). If such a word exists, then R will decode $y \mapsto x$. The important observation is that x is the word having been sent unless more than one error occurred under transmission. We say that a code with minimum distance 3 **corrects one error.** By the same reasoning, if $d > 2e$, then a code with minimum distance d corrects e errors.

To sum up: we want to use codes, with a given number of code words (the number of source states), with high minimum distance (in order to be able and correct many transmission errors), and with small length (in order to save transmission time).

Back to our toy example: the binary alphabet, 16 source states. The injective mapping of source states into codewords is called the **encoding**. Let us use now a code of length 7 and the following encoding:

0000	$\longrightarrow$	0000000
1000	$\longrightarrow$	1101000
0100	$\longrightarrow$	1010100
0010	$\longrightarrow$	0110010
0001	$\longrightarrow$	1110001
1100	$\longrightarrow$	0111100
1010	$\longrightarrow$	1011010
1001	$\longrightarrow$	0011001
0110	$\longrightarrow$	1100110
0101	$\longrightarrow$	0100101
0011	$\longrightarrow$	1000011
1110	$\longrightarrow$	0001110
1101	$\longrightarrow$	1001101
1011	$\longrightarrow$	0101011
0111	$\longrightarrow$	0010111
1111	$\longrightarrow$	1111111

It is easy to check that this code has minimum distance $d = 3$, hence corrects one error. The probability of correct decoding of a source state (there are 16 sourse states and as many codewords) is therefore $\geq (1-p)^7 + 7p(1-p)^6 = (1-p)^6(1+6p) \sim 1 - 21p^2 + 70p^3 \pm \ldots$. In the case of the repetition code the corresponding probability was $(1-p)^8(1+2p)^4 = (1-8p+28p^2+\ldots)(1+8p+24p^2+\ldots) = 1-12p^2\ldots$. We see that these probabilities have the same order of magnitude (the linear term in p vanishes in both cases), the probability of correct decoding being just a little bit lower in the case of our new code. However, the transmission of a source state is by a factor 7/12 faster with the new code as compared to the repetition code. This is a considerable gain.

The code introduced above has the property that the sum of any two code words (coordinatewise, sum mod 2, also called XORing) is a code word. In other words, the 16 codewords form a 4-dimensional subspace of the space $\mathbb{F}_2^7$ of all binary 7-tuples. As the underlying field is the field $\mathbb{F}_2$ of 2 elements, one speaks of a binary linear code. In the following chapter we will give an introduction to the construction of finite fields. Most of our codes will be linear, according to the following definition:

Definition (1.3) (linear codes) *Let $\mathbb{F}_q$ be the field with q elements. A q-ary linear code of length n and dimension k is a linear subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$ of vector space dimension k. If its minimum distance is d, then we record the parameters of $\mathcal{C}$ as*

$$[n, k, d]_q.$$

Observe that a q-ary linear code of dimension k has q^k code words. It contains the all-0-word $0 = (0, 0, \ldots, 0)$. The code which we used above has parameters $[7, 4, 3]_2$. It is known as a binary **Hamming code.** We will come back to Hamming codes in general in Chapter 7.
For fixed q we wish to construct codes $[n, k, d]$ with large d, large k and small n. In Chapter 4 we will give bounds on the possible parameters of codes, both in the general and in the linear case. Recall that a code with minimum distance d allows the correction of e transmission errors, when $2e < d$.
A last simple but important remark: the computation of the minimum distance simplifies in the linear case. In fact, we have $d(x, y) = d(x - z, y - z)$ from the definition of Hamming distance. If x, y are code words of a linear code, we get $d(x, y) = d(0, y - x)$ and $y - x$ is a code word, because of linearity. Define the **weight** $wt(x)$ as the number of coordinates with a nonzero entry, in other words $wt(x) = d(x, 0)$. Then the minimum distance of a linear code equals the minimum weight of its nonzero code words. Take as example the Hamming code $[7, 4, 3]_2$. As all its nonzero code words have weight ≥ 3 we have checked that its minimum distance is indeed $d = 3$.

1. Codes can be used to correct errors in the **transmission of messages** over **noisy channels.** In the sequel we will encounter other applications of codes, but this application was the first one historically.

2. We have seen an example of a code capable of correcting one error.

3. The **Hamming metric** underlies all constructions.

4. The **minimum distance** is one of the two most important parameters of a code. Its partner, the **strength**, is related to **orthogonal arrays** and will be introduced in Chapter 5.

5. The idea of using **linear codes** will permit us to bring linear algebra into play.

1.1 Problems

1. Prove that the Hamming distance defines a metric.

2. Compute the probabilities of reliable transmission of source states for the BSC with symbol error probability p for the three codes considered in this chapter (a) no coding, length 4, b) repetition code, length 12, c) Hamming code length 7), for $p = 0.1$, for $p = 0.01$ and for $p = 0.001$.

Chapter 2

Finite fields

Let p be a prime number. Then $\mathbb{Z}/p\mathbb{Z}$ is a finite field. We denote it by $\mathbb{F}_p$. Let F be any finite field. Denote the sum of n copies of 1 by $n \cdot 1 \in F$. As F is finite the $n \cdot 1 \in F$ cannot all be different. So there must be some $m < n$ such that $n \cdot 1 = m \cdot 1$. It follows $(n - m) \cdot 1 = 0$. Denote by a the smallest natural number such that $a \cdot 1 = 0$. As F has no zero divisors we conclude that $a = p$ must be prime. It follows from the minimality that p is the only prime with this property and that $n \cdot 1 = 0$ if and only if n is a multiple of p. We see that the $i \cdot 1, i = 0, 1, \dots p - 1$ form a subfield of F, which is isomorphic to $\mathbb{F}_p$. We call p the **characteristic** of F and $\mathbb{F}_p$, the subfield of F generated by 1, its **prime field.** So every finite field F may be described as an extension of its prime field $\mathbb{F}_p$. As F is by definition a vector space over $\mathbb{F}_p$ its number of elements is p^n for some n.

Theorem (2.1) *Every finite field has p^n elements for some prime p. The subfield generated by the element 1 is $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$.*

In order to generate finite fields we may use irreducible polynomials. So let $f(X) \in \mathbb{F}_p[X]$ be an irreducible polynomial of degree n. Take $f(X)$ to be monic (its leading coefficient is 1), so that $f(X) = X^n + a_{n-1}X^{n-1} + \dots + a_1X + a_0$. We claim that $F = \mathbb{F}_p[X]/(f(X))$, the factor ring of the polynomial ring over the ideal generated by $f(X)$, is a field with p^n elements:
denote by x the image of X mod the ideal $(f(X))$. Remark that $(f(X))$ simply is the set of all polynomials, which are divisible by $f(X)$. At first we see that F is a vector space of dimension n over $\mathbb{F}_p$, so $\mid F \mid = p^n$. The

elements of F can be uniquely represented in the form $u = \sum_{i=0}^{n-1} c_i x^i$. In fact, as $x^n = -a_{n-1}x^{n-1} - \ldots - a_1 x - a_0$, every element of F has this form. On the other hand, the $x^i, i = 0, 1, \ldots, n-1$ are linearly independent as otherwise $f(X)$ would divide a nonzero polynomial of degree $< n$, which is impossible. So every element of F can be written in a unique way as a polynomial of degree $< n$ with coefficients in $\mathbb{F}_p$. Assume $g(x), h(x)$ are such polynomials and $g(x)h(x) = 0$. This means that $f(X)$ divides $g(X)h(X)$. As $f(X)$ is irreducible it must divide either $g(X)$ or $h(X)$. Thus either $g(x) = 0$ or $h(x) = 0$. We have shown that F has no zero divisors. It remains to show that every nonzero element of F has a multiplicative inverse. So let $g(X)$ a nonzero polynomial of degree $< n$. As $f(X)$ is irreducible, it must be coprime to $g(X)$. By Euclidean algorithm we can find polynomials such that $1 = g(X)h(X) + f(X)l(X)$. If we read this mod $(f(X))$ we get $1 = g(x)h(x)$ and we have found the multiplicative inverse. In fact, these arguments are valid for any ground field. We have shown the following:

Theorem (2.2) *Let K be a field and $f(X)$ an irreducible monic polynomial of degree n over K. Then $F = K[X]/(f(X))$ is a field. It contains K as a subfield and is a vector space of dimension n over K. If x denotes the image of X mod $f(X)$, then the $x^i, i = 0, 1, \ldots, n-1$ form a basis of F as a K-vector space. We call n the* **degree** *of F over K.*

We accept from field theory the fact that an **algebraic closure** always exists and is uniquely determined. Denote by $\overline{\mathbb{F}}_p$ a fixed algebraic closure of $\mathbb{F}_p$. Recall that this means two things: firstly every element $a \in \overline{\mathbb{F}}_p$ is **algebraic** over $\mathbb{F}_p$, that is it satisfies a polynomial equation with coefficients in $\mathbb{F}_p$. Secondly, $\overline{\mathbb{F}}_p$ is **algebraically closed,** equivalently every polynomial with coefficients in $\overline{\mathbb{F}}_p$ splits into linear factors over that same field. Consider the polynomial $X^{p^n} - X$. Assume a field with p^n elements exists. As it is finite it must be algebraic over $\mathbb{F}_p$, so it can be considered as a subfield of $\overline{\mathbb{F}}_p$. As the multiplicative group of this field has order $p^n - 1$, each nonzero element u satisfies $u^{p^{n-1}} = 1$. Thus every element of a field of p^n elements is a root of our polynomial. We see that a field of order p^n is uniquely determined as a subfield of the algebraic closure, if it exists. On the other hand, the polynomial $X^{p^n} - X$ has p^n different roots. It suffices to check that these do form a field. As we are working inside a field it is sufficient to prove that sums, products and multiplicative inverses of roots are roots. For products and inverses this is obvious. For sums this is a consequence of the following Lemma:

Lemma (2.1) (Frobenius automorphism) *Let F be a field of characteristic p. Then the mapping σ, where $\sigma(x) = x^p$, is a field automorphism from F onto the field F^p of p-th powers. In the case of a finite field we have $F^p = F$. The fixed field of σ is $\mathbb{F}_p$.*

Proof: It suffices to prove that σ is linear with respect to addition and multiplication. In the case of multiplication this is obvious. Consider addition: obviously the binomial theorem applies, hence $(x+y)^p = \sum_{i=0}^{p} \binom{p}{i} x^i y^{p-i}$. Here the binomial coefficients are field elements, hence in $\mathbb{F}_p$. We see that $\binom{p}{i}$ is divisible by p and hence $= 0$ unless $i = 0$ or $i = p$. Hence the sum simplifies giving the desired result: $(x+y)^p = x^p + y^p$.$\blacksquare$

We conclude that our field of p^n elements exists and is uniquely determined.

Theorem (2.3) *For every prime p and natural number n there is a field with p^n elements. Moreover a fixed algebraic closure $\overline{\mathbb{F}_p}$ contains precisely one subfield with p^n elements, consisting of the roots of the polynomial $X^{p^n} - X$. We denote this field by $\mathbb{F}_{p^n}$.*

Once a field $\mathbb{F}_{q^n}$ is constructed we can go through the same process and construct fields of q^{nk} elements as extensions of $\mathbb{F}_{q^n}$, for every k. As we saw that these fields are uniquely determined we conclude that $\mathbb{F}_{q^n} \subset \mathbb{F}_{q^m}$ provided n divides m. On the other hand, assume $\mathbb{F}_{q^n} \subset \mathbb{F}_{q^m}$. Then the big field is a vector space over the small field. It follows that q^m must be a power of q^n, so n divides m. We have seen the following:

Theorem (2.4) *We have $\mathbb{F}_{q^n} \subseteq \mathbb{F}_{q^m}$ if and only if n divides m.*

Theorem (2.5) *The multiplicative group of a finite field is cyclic.*

Proof: Let $F = \mathbb{F}_{p^n}$. The multiplicative group of F is abelian of order $p^n - 1$. Assume it is not cyclic. Then for some prime l there must be a subgroup $Z_l \times Z_l$, an elementary abelian subgroup of order l. This is impossible as it would yield l^2 roots of the polynomial $X^l = 1$.■

To sum up: we have found, for every $q = p^m$, a uniquely determined field $\mathbb{F}_q$ of q elements. Consider its extension field $\mathbb{F}_{q^n}$.

Lemma (2.2) *Consider the field extension $\mathbb{F}_{q^n} \supset \mathbb{F}_q$ for some prime-power q. The mapping σ, where $\sigma(x) = x^q$, is a field automorphism of $\mathbb{F}_{q^n}$ over $\mathbb{F}_q$, this last term meaning that each element of the ground field is fixed under σ. This powers of σ form a group of automorphisms of order n. We call this group the Galois group $G(\mathbb{F}_{q^n}|\mathbb{F}_q)$.*

Proof: Our σ is a power of the Frobenius automorphism introduced in (2.1), so it certainly is a field automorphism of $\mathbb{F}_{q^n}$. As the elements of $\mathbb{F}_q$ satisfy $x^q = x$ we see that each element of $\mathbb{F}_q$ is fixed by σ. For the same reason we see that σ^n acts as the identity mapping on $\mathbb{F}_{q^n}$, and this is not the case for any smaller power of σ.■

Definition (2.1) (trace) *Let σ be the generator of the Galois group of $\mathbb{F}_{q^n}|\mathbb{F}_q$ as introduced in Lemma (2.2). Then the* **trace** *$tr : \mathbb{F}_{q^n} \longrightarrow \mathbb{F}_q$ is defined by*

$$tr(x) = \sum_{i=0}^{n-1} \sigma^i(x) \tag{2.1}$$

So $tr(x)$ is defined as the sum of the images of x under the elements of the Galois group. It is obvious that tr is an $\mathbb{F}_q$-linear mapping. As another application of σ permutes the elements of the Galois group we see that $\sigma(tr(x)) = tr(x)$. It follows $tr(x) \in \mathbb{F}_q$, as stated in (2.1). Moreover tr is not identically 0, as otherwise a polynomial of degree q^{n-1} would have q^n roots. We conclude that the kernel of tr is a hyperplane (an $(n-1)$-dimensional subspace) of $\mathbb{F}_{q^n}$, seen as a vector space over $\mathbb{F}_q$). For every $x \in \mathbb{F}_{q^n}$ the mapping $y \longrightarrow tr(xy)$ is a linear functional. As the space of linear functionals of an n-dimensional vector space clearly has dimension n we can describe every linear functional in this way:

Proposition (2.1) *Let F be an n-dimensional vector space over $\mathbb{F}_q$. Impose the structure of $\mathbb{F}_{q^n}$ on F. Then the linear functionals of F are in bijection with the elements of F, each $x \in F$ yielding the linear functional $y \longrightarrow tr(xy)$.*

We also see that dual bases always exist. In fact, let $v_1, v_2, \ldots, v_n$ be a basis of $\mathbb{F}_q^n$. Consider the linear functionals ϕ_i, where $\phi_i(\sum_j \alpha_j v_j) = \alpha_i$. Choose $x_i \in \mathbb{F}_{q^n}$ such that $tr(x_i y) = \phi_i(y)$. Then $tr(x_i v_j) = \delta_{ij}$. We have seen the following:

Theorem (2.6) *Let F be an n-dimensional vector space over $\mathbb{F}_q$. Impose the structure of $\mathbb{F}_{q^n}$ on F. For every basis $v_1, v_2, \ldots, v_n$ of $F|\mathbb{F}_q$ there exists another basis $x_1, x_2, \ldots, x_n$ (the* **trace-dual basis***) such that*

$$tr(x_i v_j) = \delta_{ij}.$$

2.1 Individual finite fields

It is in general handy to fix the multiplicative structure of the field (simply a cyclic group, as we know) and to determine the additive structure afterwards, using the irreducible polynomial. Write $\mathbb{F}_q = \{0\} \cup \{\epsilon^i \mid i = 0, 1, \ldots, q-1\}$. We know that the field is independent of the irreducible polynomial $f(X)$ chosen to describe it. However, there are good and bad choices. It is for example obviously advantageous to choose a polynomial of **maximal exponent** in the sense that the image of the indeterminate X in $F = K[X]/(f(X))$ generates the multiplicative group of F (see Theorem (2.2)). We will always do this. The addition in $\mathbb{F}_q$ will be completely known once the $1 + \epsilon^i$ are known for all i.

2.1.1 $\mathbb{F}_4$

The only irreducible $\mathbb{F}_2$-polynomial of degree 2 is $f(X) = X^2 + X + 1$. It follows $1 + \epsilon = \epsilon^2$. This determines the addition. For example $1 + \epsilon^2 = \epsilon, \epsilon + \epsilon^2 = \epsilon(1 + \epsilon) = \epsilon\epsilon^2 = 1$.

2.1.2 $\mathbb{F}_8$

We choose $f(X) = X^3 + X^2 + 1$, hence $1 + \epsilon^2 + \epsilon^3 = 0$. It is easy to prove that $f(X)$ is indeed irreducible over $\mathbb{F}_2$ (see the Problems section). Further $1 + \epsilon^4 = (1 + \epsilon^2)^2 = (\epsilon^3)^2 = \epsilon^6$, and then necessarily $1 + \epsilon = \epsilon^5$. We know that these relations:

$$1+\epsilon^2+\epsilon^3=0,\ 1+\epsilon^4=\epsilon^6,\ \text{and } 1+\epsilon=\epsilon^5 \tag{2.2}$$

determine the field structure completely. As an example, $\epsilon^3+\epsilon^4=\epsilon^3(1+\epsilon)=\epsilon^3\epsilon^5=\epsilon$.

2.1.3 $I\!F_{16}$

Here we have two choices. We can represent $I\!F_{16}$ either as an extension of degree 4 of $I\!F_2$ or as an extension of degree 2 of $I\!F_4$. Let us consider $I\!F_4$ as the ground field. Write $I\!F_4=\{0,1,\omega,\omega^2\}$. Consider the irreducible polynomial $f(X)=X^2+X+\omega$. This leads to the relation $\epsilon^2+\epsilon+\omega=0$. In order to see that $f(X)$ has maximal exponent write the elements of $I\!F_{16}$ as $\alpha\epsilon+\beta$, where $\alpha,\beta\in I\!F_4$. It follows $\epsilon^3=(\epsilon+\omega)\epsilon=\epsilon^2+\epsilon\omega=\epsilon+\omega+\epsilon\omega=\epsilon\omega^2+\omega$. Proceeding in the same way we get $\epsilon^4=\epsilon+1,\epsilon^5=\epsilon^2+\epsilon=\omega$. As ω has order 3, it ic clear that ϵ has order 15, thus $f(X)$ has maximal exponent. The remaining powers of ϵ are obtained by observing $\epsilon^{5+i}=\omega\epsilon^i,\epsilon^{10+i}=\omega^2\epsilon^i$. The additive structure is already determined:

$1+\epsilon+\epsilon^4=0$	$1+\epsilon^2+\epsilon^8=0$	$1+\epsilon^3+\epsilon^{14}=0$
$1+\epsilon^6+\epsilon^{13}=0$	$1+\epsilon^7+\epsilon^9=0$	$1+\epsilon^{11}+\epsilon^{12}=0$

Let $tr=tr:I\!F_{16}\longrightarrow I\!F_4$ be the trace. With the table above it is easy to calculate the trace: $tr(\epsilon)=\epsilon+\epsilon^4=\epsilon(1+\epsilon^3)=\epsilon\epsilon^{14}=1$, and $tr(\epsilon^3)=\epsilon^3+\epsilon^{12}=\epsilon^3(1+\epsilon^9)=\epsilon^3\epsilon^7=\epsilon^{10}=\omega^2$.

2.1.4 $I\!F_9$

Take X^2-X-1 as irreducible polynomial. This leads to the relation $\epsilon^2=\epsilon+1$. Then $\epsilon^3=-\epsilon+1,\epsilon^4=-1$, as it should be.

2.1.5 $I\!F_{27}$

Our polynomial is X^3-X-1. We obtain

$$
\begin{array}{|ccc|}
\hline
\epsilon^3 & = & \epsilon + 1 \\
\epsilon^4 & = & \epsilon^2 + \epsilon \\
\epsilon^5 & = & \epsilon^2 + \epsilon + 1 \\
\epsilon^6 & = & \epsilon^2 - \epsilon + 1 \\
\epsilon^7 & = & -\epsilon^2 - \epsilon + 1 \\
\epsilon^8 & = & -\epsilon^2 - 1 \\
\epsilon^9 & = & \epsilon - 1 \\
\epsilon^{10} & = & \epsilon^2 - \epsilon \\
\epsilon^{11} & = & -\epsilon^2 + \epsilon + 1 \\
\epsilon^{12} & = & \epsilon^2 - 1 \\
\epsilon^{13} & = & -1 \\
\hline
\end{array}
$$

We have $tr(1) = 0$ and $tr(\epsilon) = \epsilon + \epsilon^3 + \epsilon^9 = \epsilon + (\epsilon + 1) + (\epsilon - 1) = 0$. The kernel of the trace is therefore the subspace generated by 1 and ϵ.

1. For every prime-power q there is a finite field $\mathbb{F}_q$ with q elements.
2. If p is a prime, then $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$.
3. If $q = p^f$, then $\mathbb{F}_q$ can be constructed as an extension of $\mathbb{F}_p$, with the help of an irreducible polynomial.
4. Some small finite fields have been given in detail.
5. Among the most important notions concerning finite fields are the **trace** and the **Frobenius automorphism.**
6. In later chapters the trace $tr : \mathbb{F}_{p^f} \longrightarrow \mathbb{F}_p$ will be used to construct p-ary codes out of q-ary codes.

2.2 Problems

1. Prove that the polynomial $X^3 + X^2 + 1$ is irreducible over $\mathbb{F}_2$.
2. Find a self-dual basis of $\mathbb{F}_8 \mid \mathbb{F}_2$ (see Theorem (2.6)).
3. Determine the product of all nonzero elements of $\mathbb{F}_q$.

4. Determine the sum of all elements of $\mathbb{F}_q$.
5. Prove the following: if $a, b \in \mathbb{F}_9$ are nonsquares and $a \neq \pm b$, then $a + b$ and $a - b$ are squares.
   Prove that every element of $\mathbb{F}_9$ is an $\mathbb{F}_3$-linear combination of two nonsquares.

Chapter 3

The Reed-Solomon codes

We describe an array whose rows are the code words. Let the rows be parametrized by the polynomials $p(X)$ with coefficients in $\mathbb{F}_q$, of degree $< k$. The columns are parametrized by the elements $u \in \mathbb{F}_q$. The entry in row $p(X)$ and column u is the evaluation $p(u)$. Obviously this defines a linear q-ary code of length q. Assume $k < q$. The number of roots of a nonzero polynomial of degree $< k$ is $\leq k-1$. It follows that the minimum distance is $\geq q-(k-1) = q-k+1$. As there are polynomials of degree $k-1$ with $k-1$ different roots we have equality.

Definition (3.1) (Reed-Solomon codes) *Let $k < q$. The words of the* **Reed-Solomon code** $\mathcal{RS}(k,q)$ *of dimension k over $\mathbb{F}_q$ are parametrized by the polynomials $p(X) \in \mathbb{F}_q[X]$ of degree $< k$, the coordinates by $u \in \mathbb{F}_q$. The corresponding entry is $p(u)$. The code $\mathcal{RS}(k,q)$ has length q, dimension k and minimum distance $q-k+1$.*

This gives us a large class of important codes. Here is the Reed Solomon code $\mathcal{RS}(2,4)$, where we write $\mathbb{F}_4 = \{0,1,\omega,\omega^2\}$.

0	0	0	0
1	1	1	1
ω	ω	ω	ω
ω^2	ω^2	ω^2	ω^2
0	1	ω	ω^2
1	0	ω^2	ω
ω	ω^2	0	1
ω^2	ω	1	0
0	ω	ω^2	1
1	ω^2	ω	0
ω	0	1	ω^2
ω^2	1	0	ω
0	ω^2	1	ω
1	ω	0	ω^2
ω	1	ω^2	0
ω^2	0	ω	1

Here the columns are indexed by $0, 1, \omega, \omega^2$ (in this order), the row sections correspond to constant polynomials, $X + c, \omega X + c, \omega^2 X + c$. The fifth row represents polynomial X. We also see that the minimum distance of $\mathcal{RS}(2, 4)$ is indeed $d = 3$. We will see in the next chapter that the parameters $[q, k, q - k + 1]_q$ of the Reed Solomon codes are very good.

3.1 Problems

1. Write all code words of $RS(2, 5)$.

2. Write all code words of $RS(3, 4)$.

Chapter 4

Bounds on codes

We begin our study of codes with some bounds on the parameters of codes. Familiarity with these bounds will come in handy so that we will be able to appreciate the codes to be constructed in the next chapter.

Some bounds are very easily obtained. Here is our first example: Fix some $n-(d-1)$ coordinates of a code. After projection to these coordinates different codewords remain different. This is true because of the definition of minimum distance. We have proved the following bound:

Theorem (4.1) (Singleton bound) *Let $\mathcal{C}$ be a v-ary code of length n. Then the following holds:*

$$\mid \mathcal{C} \mid \leq v^{n-d+1}.$$

Codes satisfying this bound with equality are traditionally called **MDS-codes**. *Here the letters* **MDS** *stand for* **maximum distance separable.**

In the linear case this simplifies to the inequality $d+k \leq n+1$. We see that the **Reed Solomon codes** $RS(k,q)$ **are MDS.**

The next bound is equally trivial. Consider the space $\mathcal{A}^n$ of vectors (n-tuples) of length n, with the Hamming metric. This is called the **Hamming space.** The number of vectors at distance $\leq i$ from a given vector clearly is $V_v(i,n)=\sum_{j=0}^{i}\binom{n}{j}(v-1)^j$.

Definition (4.1) *The* **volume of the ball of radius** i *in Hamming space is*

$$V_v(i,n)=\sum_{j=0}^{i}\binom{n}{j}(v-1)^j \tag{4.1}$$

Put $e=\lfloor(d-1)/2\rfloor$. It follows from the definition of the minimum distance that the balls of radius e around the codewords do not overlap. The sum of their volumes can therefore not be larger than v^n, the number of all vectors in Hamming space. This is our second bound:

Theorem (4.2) (sphere packing bound) *Each v-ary code $\mathcal{C}$ of length n and minimum distance d satisfies*

$$\mid \mathcal{C}\mid \cdot V_v(e,n)\leq v^n.$$

Here $e=\lfloor(d-1)/2\rfloor$.

Codes satisfying the sphere packing bound with equality are known as **perfect codes.** The Hamming code $[7,4,3]_2$ is indeed perfect. It will turn out that the Hamming codes to be constructed in Chapter 7 are all perfect.

Another important bound on codes follows from the elementary technique known as **quadratic counting:**

Theorem (4.3) (Plotkin bound) *Let $\mathcal{C}$ be a v-ary code of length n and minimum distance d. Assume moreover $d>\frac{v-1}{v}\cdot n$. Then the following holds:*

$$\mid \mathcal{C}\mid \leq d/(d-\frac{v-1}{v}n).$$

Proof: We count the triples (x,y,j), where x and y are different codewords and j is a coordinate, where x and y differ. Put $M=\mid\mathcal{C}\mid$. It follows from the definition of d that the number of such triples is $\geq M(M-1)d$. Let us count each coordinate seperately now. So fix a coordinate j and let m_i denote the number of codewords having entry i in the given coordinate. We have $\sum_i m_i=M$ and the contribution of the coordinate to our counting problem $\sum_i m_i(M-m_i)=M^2-\sum_i m_i^2$. We need an upper bound on this expression, hence a lower bound on $\sum_i m_i^2$. This bound follows from the classical **Cauchy-Schwartz inequality.** An easy proof is sketched in the

Problems section. The inequality states the following, which is rather clear intuitively: if the sum M of v nonnegative numbers m_i is fixed, then the sum $\sum_i m_i^2$ of their squares is minimized by choosing all m_i to be equal, hence $= M/v$. In that case $\sum_i m_i^2 = M^2/v$. It follows that in our situation the contribution of each coordinate is $\leq M^2 - M^2/v = M^2(v-1)/v$.
Taking both bounds together we get $M(M-1)d \leq nM^2(v-1)/v$. Solving for M the Theorem is obtained. ∎

Our next bound is applicable to linear codes only. So let $\mathcal{C}$ be a linear q-ary code $[n,k,d]$. Choose a codeword $x \in \mathcal{C}$ of weight d. We can assume without restriction that x has entry 1 in the first d coordinates, 0 otherwise. We construct a code, which is known as the **residual code** of $\mathcal{C}$ with respect to x, as follows: consider a complement of the 1-dimensional subspace $\mathbb{F}_q \cdot x$ in $\mathcal{C}$, project its vectors onto the last $n-d$ coordinates.
This residual code $\mathcal{D}$ has dimension $k-1$ (if its dimension were smaller, then $\mathcal{C}$ would have to contain a word $\neq 0$ of weight $< d$). What can we say about the minimum distance d' of $\mathcal{D}$? Let $y \in \mathcal{C}$ be a codeword, whose projection has weight i in $\mathcal{D}$. In the first d coordinates y must have at least $d-i$ entries $\neq 0$. It follows that there must be at least $(d-i)/(q-1)$ coordinates in the first segment, where y has the same nonzero entry. Because of linearity we can assume that this entry is $= 1$. Consider $y-x$. We have $0 \neq y-x \in \mathcal{C}$, thus $d \leq wt(y-x) \leq i + d - \frac{d-i}{q-1}$. Solving this for i we obtain $i \geq d/q$. We have shown $d' \geq \lceil d/q \rceil$. By iterating this process the Griesmer bound is obtained.

Theorem (4.4) (Griesmer bound) *Let $\mathcal{C}$ be a q-ary linear code with parameters $[n,k,d]_q$. Then there is a q-ary code $[n-d, k-1, \geq \lceil d/q \rceil]_q$ and the following holds:*

$$n \geq \sum_{i=0}^{k-1} \lceil d/q^i \rceil.$$

Finally observe that we do not have to know the exact minimum distance of $\mathcal{C}$ for this theorem to be true. If we know only that the minimum distance is $\geq d$, we start from a nonzero vector x of minimum weight and the Theorem still is true.

1. The basic parameters of a linear q-ary code are the **length** n, the **dimension** k and the **minimum distance** d.
2. We write the parameters $[n, k, d]_q$.
3. We have derived several inequalities:
4. The **Singleton bound** $k + d \leq n + 1$ is almost trivial.
5. The **sphere packing bound** follows from a combinatorial argument.
6. The **Plotkin bound** is valid for large minimum distances. It is derived by quadratic counting.
7. The **Griesmer bound** is related to **residual codes** (see Chapter 10). It generalizes the Singleton bound and is the only one of these bounds, which has no counterpart for nonlinear codes.

4.1 Problems

1. Let the natural number n and the positive real number S be given. Consider nonnegative numbers $x_i, i = 1, 2, \ldots, n$ such that $\sum_{i=1}^n x_i = S$. Prove: the expression $\sum_{i=1}^n x_i^2$ is minimized by the choice $x_i = S/n$. Determine this minimum.
   Hint: Let some n-tuple $(x_1, x_2, \ldots, x_n)$ be given such that not all the x_i equal S/n. Pick i such that $x_i > S/n$ and j such that $x_j < S/n$. Show that you can replace x_i by $x_i - \epsilon$ and x_j by $x_j + \epsilon$ for some $\epsilon > 0$ and obtain a tuple with a smaller sum of squares.
2. Let $\mathcal{C}$ be a q-ary code $[n, k, d]_q$. Prove: if a code $[n-1, k, d]_q$ does not exist, then a code $[n, k, d+1]_q$ does not exist.
3. Prove that a linear code $[7, 4, 3]_2$ is perfect.
4. What is the maximum possible d of a ternary code $[11, 6, d]_3$?

Chapter 5

Duality and orthogonal arrays

Let $\mathbb{F}_q^n$ be the space of n-tuples with entries in $\mathbb{F}_q$. We consider the usual dot product on $\mathbb{F}_q^n$:

Definition (5.1) (dot product) *The dot product is defined on $\mathbb{F}_q^n$ by*

$$(x_1, x_2, \ldots, x_n) \cdot (y_1, y_2, \ldots, y_n) = \sum_{i=1}^{n} x_i y_i \tag{5.1}$$

Let $x, y \in \mathbb{F}_q^n$. We say that x and y are **orthogonal** ($x \perp y$) if $x \cdot y = 0$. It is obvious that for every subset $U \subseteq \mathbb{F}_q^n$ the set

$$U^\perp = \{x \mid x \in \mathbb{F}_q^n, x \cdot u = 0 \text{ for all } u \in U\}$$

is a subspace of $\mathbb{F}_q^n$. The following rules hold for all $x, y, z \in \mathbb{F}_q^n$ and $\lambda \in \mathbb{F}_q$:

$$x \cdot y = y \cdot x$$

$$(x + y) \cdot z = x \cdot z + y \cdot z$$

$$(\lambda x) \cdot y = x \cdot (\lambda y) = \lambda(x \cdot y).$$

This means that the dot-product is a **symmetric bilinear form.**

Lemma (5.1) *The only vector $x \in \mathbb{F}_q^n$ satisfying $x^\perp = \mathbb{F}_q^n$ is the 0-vector $x = 0$.*

The proof is trivial. Lemma (5.1) says that the dot product is what is called **non-degenerate.**

Lemma (5.2) *Let $U \subseteq \mathbb{F}_q^n$ be a linear subspace. Then*

$$dim(U) + dim(U^{\perp}) = n.$$

Proof: Put $dim(U) = k$. The statement is obvious when $k = 0$. In case $k = n$ the statement is equivalent to the non-degeneracy of the dot product. It is clear that the statement holds when $k = 1$. In the general case we embed U in a chain

$$0 \subset U_1 \subset \ldots \subset U_k = u \ldots \subset U_n = \mathbb{F}_q^n.$$

Taking duals we obtain

$$\mathbb{F}_q^n \supset U_1^{\perp} \supset \ldots \supset U^{\perp} \ldots \supset U_n^{\perp} = 0.$$

As $U_{i+1}^{\perp}$ is the intersection of $U_i^{\perp}$ with a hyperplane (a space of dimension $n-1$), it follows $dim(U_{i+1}^{\perp}) \geq dim(U_i^{\perp}) - 1$, by induction $dim(U_i^{\perp}) \geq n - i$. As we know that $dim(U_n^{\perp}) = 0$ (as a consequence of the non-degeneracy) we must have equality all the way.■

In particular we see that when $\mathcal{C}$ is a linear k-dimensional q-ary code of length n, then the **dual code** $\mathcal{C}^{\perp}$ has dimension $n - k$.

Lemma (5.3) *Let $U \subseteq \mathbb{F}_q^n$ be a linear subspace. Then*

$$U^{\perp\perp} = U$$

Proof: By definition $U^{\perp\perp} \supseteq U$. It follows from Lemma (5.2) that these spaces have the same dimension.■

5.1 Strength and duality

Let $\mathcal{C}$ be a linear q-ary code of dimension k. Then $\mathcal{C}$ can be described by a basis. This leads to the following definition:

Definition (5.2) (generator matrix) *A* **generator matrix** *G of a linear q-ary code is a (k, n)-matrix with entries in $\mathbb{F}_q$, such that the rows of G form a basis of the code.*

Observe that a generator matrix is far from being uniquely determined by the code, as a vectorspace has many bases. As a code $\mathcal{C}$ is uniquely determined by its dual $\mathcal{C}^\perp$ we may describe it by a generator matrix of the dual code.

Definition (5.3) (check matrix) *A generator matrix of the dual code $\mathcal{C}^\perp$ is called a* **check matrix** *for $\mathcal{C}$.*

Let H be a check matrix of $\mathcal{C}$. Denote by $c_i, i = 1, 2, \ldots, n$ the columns of H. By definition an n-tuple $(x_1, x_2, \ldots, x_n)$ is in $\mathcal{C}$ if and only if $\sum_{i+1}^n x_i c_i = 0$. This means that the elements of $\mathcal{C}$ are precisely the coefficients yielding vanishing linear combinations of the columns of H. It follows from the definition of minimum distance that every set of $d-1$ columns of H is linearly independent. This leads us to the following definition:

Definition (5.4) *The* **strength** *of a matrix H with entries in the field $\mathbb{F}_q$ is the maximum number t such that any t columns of H are linearly independent (thus the projection onto any t columns of H has full rank t).*

We have seen the following important relation between strength and minimum distance, in the case of linear codes:

Lemma (5.4) *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear subspace. Then $\mathcal{C}$ has minimum distance $\geq d$ if and only if a (every) check matrix of $\mathcal{C}$ has strength $\geq d-1$.*

We can express this in terms of the dual code instead of in terms of its generator matrix H. Clearly H has strength $\geq t$ if and only if in the restriction of $\mathcal{C}^\perp$ to any t columns any t-tuple occurs the same number of times. This leads us to the combinatorial meaning of the strength, which we cast in the following definition:

Definition (5.5) (orthogonal array) *An array with n columns and entries from a set $\mathcal{A}$ of cardinality v is an* **orthogonal array** *of strength t if in the projection onto any set of t columns each t-tuple of entries occurs the same number λ of times. We write the parameters of such an orthogonal array as*

$$OA_\lambda(t, n, v).$$

It follows that the number of columns of an $OA_\lambda(t,n,v)$ is λv^t. Observe that we do not demand in general that the rows of an OA represent different n-tuples. It is clear that an OA of strength $t>1$ is also an OA of strength $t-1$. We reformulate what we have seen above:

Theorem (5.1) *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear subspace. Then the minimum distance of $\mathcal{C}$ and the strength of $\mathcal{C}^\perp$ are related by*

$$d(\mathcal{C}) = t(\mathcal{C}^\perp) + 1.$$

We would like to call this (trivial) Theorem the **Principle of duality for linear codes.** It says that the minimum distance of a linear code is precisely one larger than the strength of its dual. Observe that any set of n-tuples has a minimum distance and a strength. One speaks of codes if one is primarily interested in the minimum distance, of orthogonal arrays if the strength is in the center of the interest. Here is a particularly simple class of examples:

Definition (5.6) (sum zero-code) *Let G be a commutative group of order v (written additively). The array with n columns, whose rows are all the n-tuples with entries in G, whose entries have sum 0, is the* **sum zero code.** *It is an orthogonal array with parameters*

$$OA_1(n-1,n,v)$$

and a code with minimum distance 2.

If $v=q$ is a prime-power and G is the additive group of $\mathbb{F}_q$, then the sum zero-code is a linear q-ary code. The sum zero-codes, seen as orthogonal arrays, correspond to a particularly simple class of codes, the repetition codes:

Definition (5.7) (repetition code) *The* **repetition code** *of length n over alphabet $\mathcal{A}$ consists of all n-tuples $(g,g,\ldots,g)$ with identical entries. If $\mathcal{A}=\mathbb{F}_q$, then the repetition code is the 1-dimensional linear code generated by the all-1-word* $(1,1,\ldots,1)$.

We leave it to the reader to check that the linear repetition code and the linear sum zero-code are orthogonal to each other (see the Problems section).

Example (5.1) (RS(2,4)) *We illustrate with the Reed-Solomon code* $\mathcal{RS}(2,4)$ *as given in Chapter 3. This code has 16 codewords (the rows of the array). We can check that any two codewords agree in at most one coordinate. In other words the minimum distance is* $d=3$. *The strength cannot exceed 2. We can check that when we run our fingers down any two columns, each pair of entries occurs precisely once. This means that the strength is indeed two. As an orthogonal array the parameters are therefore* $OA_1(2,4,4)$.
We can also check that the codewords (rows) form a vector space. For example, the sum of row 5 and row 6 is row 2, and when we multiply row 8 by ω *we obtain row 10. The parameters as a linear code are therefore* $[4,2,3]_4$.

More generally $\mathcal{RS}(k,q)$ is an $OA_1(k,q,q)$. This follows from Lagrange interpolation (see Theorem (16.2)). We shall see later that from the point of view of applications the strength is at least as important as the minimum distance. As a first hint in this direction we give a probabilistic interpretation of orthogonal arrays:

5.2 A probabilistic interpretation of orthogonal arrays

We define an elementary notion of probability theory:

Definition (5.8) (probability space) *Let* Ω *be a finite set. For every* $x\in\Omega$ *let a nonnegative number* p_x *be given such that*

$$\sum_{x\in\Omega} p_x = 1.$$

We see $p:\Omega\longrightarrow\mathbb{R}$ *as a mapping, where* $p(x)=p_x$. *Then* (Ω,p) *is a* **(finite) probability space.** *Call* p_x *the* **probability** *of* x.

Consider an array with n columns and b rows, where the entries are from a v-element set $\mathcal{A}$. Identify the rows with elements of a probability space Ω (of cardinality b) with uniform distribution, so that each row has probability $1/b$. Interpret each column as a function $f:\Omega\longrightarrow\mathcal{A}$. Functions defined on probability spaces are also known under the name of **random variables.** So we interpret our columns as a family of n random variables defined on Ω.

Obviously we consider here only finite probability spaces and discrete random variables (those whose values are chosen in a finite set). We make the following definitions:

Definition (5.9) *A discrete random variable is* **uniformly distributed** *if each of its values occurs with the same probability. A family $f_1, \ldots, f_t$ of discrete random variables defined on Ω is* **statistically independent** *if for every choice of $u_i \in f_i(\Omega)$ the following holds:*
the probability that $f_i(x) = u_i, i = 1, 2, \ldots, t$ equals the product $\prod_{i=1}^{t} Prob(f_i(x) = u_i)$ over the individual probabilities.

Theorem (5.2) *Let Ω be a probability space of cardinality b, with uniform distribution, let E be a v-set and $\mathcal{F}$ a family of n uniformly distributed random variables $f : \Omega \longrightarrow E$. Then the following are equivalent:*

- *each subset of $\leq t$ elements of $\mathcal{F}$ is statistically independent.*
- *The array whose rows are indexed by Ω, whose columns are indexed by $\mathcal{F}$ and with entry $f(x)$ in row x, column f is an orthogonal array of strength t. The parameters of this OA are then necessarily $OA_{b/v^t}(t, n, v)$.*

This is really trivial. The fact that each f is uniformly distributed means that the array is an OA of strength 1. The statistical independence of t random variables means that in each subarray corresponding to t columns each t-tuple of entries occurs equally often. This comes down to the definition of an OA of strength t.
Consider the OA of strength 2 as given in Example (5.1). We consider the rows to form a probability space Ω of 16 elements, where each row has probability 1/16. Each column is considered as a random variable with values in $\mathbb{F}_4$. These random variables are uniformly distributed as each column contains each entry the same number of times (4 times). Any two of our four random variables are statistically independent.

1. The **dual** $\mathcal{C}^\perp$ of a linear code $\mathcal{C}$ is defined with respect to the usual **dot product.** $\mathcal{C}$ and $\mathcal{C}^\perp$ have complementary dimensions.
2. A **generator matrix** of $\mathcal{C}$ has rows forming a basis of $\mathcal{C}$.
3. A **check matrix** of $\mathcal{C}$ is a generator matrix of $\mathcal{C}^\perp$.
4. The **strength** is the most important parameter of an **orthogonal array**.
5. The **Principle of duality for linear codes** states that the minimum distance of $\mathcal{C}$ is one more than the strength of its dual.
6. We have defined two trivial classes, the **sum-zero codes** and the **repetition codes.**
7. Orthogonal arrays can equivalently be defined as families of random variables satisfying certain statistical independance conditions.

5.3 Problems

1. Prove that the dual of the linear repetition code of length n is the sum zero-code.

2. Prove the **Bose-Bush bound** for orthogonal arrays, which states that if an $OA_\lambda(2, n, v)$ exists, then $\lambda v^2 > n(v-1)$.
   Hint: Denote the set of entries as $\{\infty\} \cup E$, where $E = \mathbb{Z}/(v-1)\mathbb{Z}$. Show that the first row of the orthogonal array A can be chosen with all entries ∞. Denote by $A+i$ the array, where $i \in E$ is added to every entry of A (observing the usual rules for calculation with infinity). Let $B = (A|A+1|\dots|A+(v-2))$, with the first row removed. This matrix has $l = \lambda v^2 - 1$ rows and $n(v-1)$ columns. Interpret the columns of B as words of a code. Apply the Plotkin bound Theorem(4.3).

Chapter 6

Weight distribution and MacWilliams formula

We may be interested not only in the minimum distance of a code, but also in which distances occur, and with what frequency. This more detailed information is contained in the distance distribution of a code.

Definition (6.1) *Let $\mathcal{C}$ be a v-ary code of length n. The* **distance distribution** *of $\mathcal{C}$ is the $(n+1)$-tuple of rational numbers $A_i, i = 0, 1, \ldots, n$, where*

$$A_i = (\#(x, y) \mid x, y \in \mathcal{C}, d(x, y) = i) / \mid \mathcal{C} \mid .$$

The **distance polynomial** *of $\mathcal{C}$ is $W_{\mathcal{C}}(X, Y) = \sum_{i=0}^{n} A_i X^{n-i} Y^i$.*

In the case of a linear code, A_i is the number of code words of weight i. Clearly always $A_0 = 1$. Further the minimum distance $d = d(\mathcal{C})$ is the smallest positive number such that $A_d \neq 0$. In the case of a linear code, the distance polynomial of the dual code is obtained by a simple polynomial transformation. This result is attributed to F.J. MacWilliams (1963). In reality this can be seen as a finite (and particularly simple) version of the old Theta transformation formula from the theory of modular forms.

Theorem (6.1) (Mc Williams formula) *Let $\mathcal{C}$ be a linear q-ary code with distance polynomial $W_{\mathcal{C}}(X, Y)$. Then the distance polynomial of the dual code is*

$$\frac{1}{\mid \mathcal{C} \mid} W_{\mathcal{A}}(X + (q-1)Y, X - Y) \tag{6.1}$$

Proof: The following probabilistic proof is taken from [12]. It is motivated by the symmetric channel without memory, where during transmission in every coordinate an error occurs with probability $\epsilon/(q-1)$ for some fixed ϵ. Interpret $x \in \mathbb{F}_q^n$ as an error pattern. Clearly the probability of a particular error pattern x is $P(x) = (\frac{\epsilon}{q-1})^{wt(x)}(1-\epsilon)^{n-wt(x)}$. This turns $\mathbb{F}_q^n$ into a probability space. As usual the probability of a subspace $S \subseteq \mathbb{F}_q^n$ is defined as $P(S) = \sum_{x \in S} P(x)$. Then $p(\mathcal{C})$ is the probability that a codeword is received when a codeword was sent. We calculate $p(\mathcal{C})$ in two ways: For one thing we certainly have

$$p(\mathcal{C}) = \sum_{i=0}^{n} A_i (1-\epsilon)^{n-i} \cdot (\frac{\epsilon}{q-1})^i = W_{\mathcal{C}}(1-\epsilon, \frac{\epsilon}{q-1}).$$

Let us calculate $p(\mathcal{C})$, in a different way, based on the dual code: Compute

$$\sum_{(x,y)} p(x) \text{ ,where } y \in \mathcal{C}^{\perp}, x \cdot y \neq 0.$$

$x \in \mathcal{C}$ yields no contribution to the sum, each $x \notin \mathcal{C}$ contributes $p(x)(q^{n-k} - q^{n-k-1})$. It follows that the sum is $(1-p(\mathcal{C}))(q^{n-k} - q^{n-k-1})$. Fix now a word $y \in \mathcal{C}^{\perp}$ of weight w. Its contribution to the sum will depend only on w. We denote it by p_w. If we denote by j the size of the intersection of the supports of y and x then $p_w = \sum_{j=1}^{w} \binom{w}{j} (1-\epsilon)^{w-j}(\epsilon/(q-1))^j \cdot f(j)$, where $f(j)$ is the number of j-tuples with nonzero coordinates whose sum is nonzero. Clearly $f(1) = q-1, f(j) = (q-1)^j - f(j-1)$. Using the geometric series or induction we obtain $f(j) = \frac{q-1}{q}((q-1)^j + (-1)^{j-1})$. Thus

$$p_w = \frac{q-1}{q}(1-\epsilon)^w \{\sum_{j=1}^{w} \binom{w}{j} (\frac{\epsilon}{1-\epsilon})^j - \sum_{j=1}^{w} (-1)^j \binom{w}{j} (\frac{\epsilon}{(1-\epsilon)(q-1)})^j\}.$$

It follows $p_w = \frac{q-1}{q}(1-\epsilon)^w\{(\frac{1}{1-\epsilon})^w - (1 - \frac{\epsilon}{(1-\epsilon)(q-1)})^w\}$, finally

$$p_w = \frac{q-1}{q}[1 - (1 - \frac{\epsilon q}{q-1})^w].$$

Summing up and solving for $p(\mathcal{C})$ we obtain

$$p(\mathcal{C}) = 1 - \frac{1}{(q-1)q^{n-k-1}} \frac{q-1}{q} \sum_{x \in \mathcal{C}^{\perp}} (1 - (1 - \frac{\epsilon q}{q-1})^{wt(x)} =$$

$$= \frac{1}{q^{n-k}} \sum_i A'_i (1 - \frac{\epsilon q}{q-1})^i = \frac{1}{\mid \mathcal{C}^\perp \mid} \cdot W_{\mathcal{C}^\perp}(1, 1 - \frac{\epsilon q}{q-1}).$$

We have seen

$$W_{\mathcal{C}}(1-\epsilon, \frac{\epsilon}{q-1}) = \frac{1}{\mid \mathcal{C}^\perp \mid} \cdot W_{\mathcal{C}^\perp}(1, 1 - \frac{\epsilon q}{q-1}).$$

Putting $X = 1-\epsilon, Y = \frac{\epsilon}{q-1}$, we get the desired formula, where $\mathcal{C}$ and its dual have changed places.■

1. The **distance distribution** of a code records which distances occur between codewords and how often. It carries more information than the number of codewords and the minimum distance.
2. The **weight** of a vector is its number of nonzero coordinates.
3. The distance distribution of a linear code agrees with its **weight distribution.**
4. The **MacWilliams formula** shows how to calculate the weight distribution of $\mathcal{C}^\perp$ out of the weight distribution of $\mathcal{C}$.

Chapter 7

The Hamming and Simplex codes

The **Simplex code** $\mathcal{S}_k(q)$ is a k-dimensional code defined by a generator matrix whose columns are representatives for all 1-dimensional subspaces of $\mathbb{F}_q^k$. Its length is therefore $\begin{bmatrix} k \\ 1 \end{bmatrix} = (q^k - 1)/(q-1)$. Let x be a nonzero codeword. We can embed x in a basis of the vector space. As the number of zero entries in any row of the matrix is $\begin{bmatrix} k-1 \\ 1 \end{bmatrix} = (q^{k-1} - 1)/(q - 1)$ it follows that every $x \neq 0$ has weight $wt(x) = \begin{bmatrix} k \\ 1 \end{bmatrix} - \begin{bmatrix} k-1 \\ 1 \end{bmatrix} = q^{k-1}$.

Definition (7.1) (Simplex codes) *The q-ary* **Simplex code** $\mathcal{S}_k(q)$ *of dimension k is a code with a generator matrix whose columns are representatives of the 1-dimensional subspaces of $\mathbb{F}_q^k$. Its length is therefore $(q^k - 1)/(q - 1)$. Each nonzero codeword has weight q^{k-1}.*

Here is a generator matrix of the binary Simplex code $\mathcal{S}_3(2)$:

1	0	0	1	1	0	1
0	1	0	1	0	1	1
0	0	1	0	1	1	1

Observe that the columns of this generator matrix are just the nonzero binary vectors of length 3.

Definition (7.2) (Hamming codes) *The Hamming code $\mathcal{H}_k(q)$ is defined as the dual of the Simplex code:*

$$\mathcal{H}_k(q) = \mathcal{S}_k(q)^{\perp} \tag{7.1}$$

It follows that the Hamming code has length $(q^k-1)/(q-1)$ and dimension $(q^k-1)/(q-1)-k$. It is also obvious that the minimum distance of the Hamming codes is 3: in fact, by definition $\mathcal{S}_k(q)$ has strength 2. Next we show that the Hamming codes meet the sphere packing bound with equality:

Theorem (7.1) *Hamming codes are perfect of minimum distance 3.*

Proof: We have seen that the minimum distance of $\mathcal{H}_k(q)$ is $d=3$. In the terminology of the sphere packing bound Theorem (4.2) we have $e=1$. The volume of the ball of radius 1 is $V_q(1,n) = 1+(q-1)n$ (see Definition (4.1)). As $n=(q^k-1)/(q-1)$ we get $V_q(1,n)=q^k$, hence $\mid \mathcal{H}_k(q) \mid \cdot V_q(1,n) = q^n$.■

The Simplex code $\mathcal{C} = \mathcal{S}_k(q)$ of length $n=(q^k-1)/(q-1)$ has weight polynomial $W(X,Y) = X^n + (q^k-1)X^{n-q^{k-1}}Y^{q^{k-1}}$. This expresses the fact that all nonzero codewords have weight q^{k-1}. The MacWilliams formula gives us the weight polynomial of the Hamming code $\mathcal{C}^{\perp} = \mathcal{H}_k(q)$. It is

$$W_{\mathcal{C}^{\perp}}(X,Y) = \frac{1}{q^k}\{(X+(q-1)Y)^n + (q^k-1)(X+(q-1)Y)^{n-q^{k-1}}(X-Y)^{q^{k-1}}\}. \tag{7.2}$$

7.1 Hamming decoding

An important question which we haven't touched at yet is the decoding algorithm. How does the receiver compute the code word that was sent, assuming not more than e errors ocurred. Here $2e<d$. In the case of the Hamming codes this is particularly easy. We illustrate with the familiar Hamming code $[7,4,3]_2$. A check matrix H has been given in the beginning of this chapter. We reproduce it here:

1	0	0	1	1	0	1
0	1	0	1	0	1	1
0	0	1	0	1	1	1

Denote the columns of H by $s_1, s_2, \ldots, s_7$. Assume the codeword x was sent. That x is a code word is equivalent to $xH^t = 0$ (in words: x is orthogonal to the rows of H). For example, take $x = (1,0,1,1,0,1,0)$. As $d = 3$ we assume at most one error ocurred. If this error ocurred in coordinate number 3, say, then

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

is the received vector. Here is how the receiver proceeds:
he or she computes $yH^t = (0,0,1)^t$. As $y = x + z$, where z is the error vector, the receiver knows that $yH^t = xH^t + zH^t = zH^t$. If zH^t is column number j of H, then the error ocurred in that column. The correct code word is obtained by adding 1 in coordinate j to x. In our example $(0,0,1) = s_3$. It follows $x = y + (0,0,1,0,0,0,0)$.
Let us consider a ternary example. A check matrix for the ternary Hamming code $\mathcal{H}_3(3)$ (equivalently: a generator matrix of $\mathcal{S}_3(3)$) is as follows:

1	0	0	1	1	0	0	1	1	1	1	1	1
0	1	0	1	2	1	1	0	0	1	1	2	2
0	0	1	0	0	1	2	1	2	1	2	1	2

In fact, we check that each of the $(3^3 - 1)/(3 - 1) = 13$ subspaces of dimension 1 is represented by a column of H. Denote these columns by $s_1, s_2, \ldots, s_{13}$. Assume $y = (1,1,0,0,2,2,0,0,1,1,2,0,0)$. We compute $yH^t = (1,1,0)^t$. This is s_4. We obtain x from y by subtracting 1 in coordinate number 4: $y \mapsto x = (1,1,0,2,2,2,0,0,1,1,2,0,0)$.
Let $y = (1,0,0,1,0,0,0,0,0,0,0,0,0)$ be received. Then $yH^t = (2,1,0)^t = 2 \cdot s_5$. In order to decode we have to subtract 2 from (equivalently: add 1 to) coordinate 5, thus $y \mapsto x = (1,0,0,1,1,0,0,0,0,0,0,0,0)$.

1. Pick a prime-power q and a natural number k.
2. Write a matrix G with k rows and $(q^k - 1)/(q-1)$ columns, whose columns are representatives for the 1-dimensional subspaces of $\mathbb{F}_q^k$.
3. Then G is a generator matrix of the Simplex code $\mathcal{S}_k(q)$ (of dimension k) and a check matrix of the Hamming code $\mathcal{H}_k(q)$.
4. $\mathcal{H}_k(q)$ is a perfect code with minimum distance 3.
5. Every nonzero codeword of $\mathcal{S}_k(q)$ has weight q^{k-1}.
6. There is a particularly easy decoding algorithm for the Hamming codes.

7.2 Problems

1. Use the binary Hamming code $[7, 4, 3]_2$. Decode the following received vectors:
   $y_1 = (1, 1, 0, 1, 1, 0, 0), y_2 = (1, 1, 1, 1, 1, 1, 1), y_3 = (1, 1, 1, 0, 0, 0, 0)$.
2. Use the ternary Hamming code $[13, 10, 3]_3$. Decode the vectors
   $y_1 = (0, 0, 0, 1, 1, 1, 2, 0, 1, 0, 0, 0, 2)$ and
   $y_2 = (0, 1, 2, 0, 0, 0, 0, 1, 2, 1, 2, 0, 0)$.

Chapter 8

How to represent a code

The most straightforward way to describe a code is the following: write an array with M rows and n columns, where the rows are just the codewords of $\mathcal{C}$. This is in many situations a very satisfactory way of describing or visualizing a code. Let us call it the **direct method.** We used the direct method to describe the Reed-Solomon codes in Chapter 3. We will mainly be concerned with linear codes. As the code then forms a vectorspace it suffices to give a basis of $\mathcal{C}$. This leads to the description of a linear code by means of a generator matrix as introduced in Definition (5.2). Our third method of describing a linear code is by means of a check matrix H. (see Definition (5.2)). This is related to the dual code. We have that x is in $\mathcal{C}$ if and only if it is orthogonal to the rows of H. In matrix terms we write this as follows:

$$x \in \mathcal{C} \text{ if and only if } xH^T = 0. \tag{8.1}$$

Here H^T denotes the transposed matrix. Observe that a check matrix H of our code $[n, k, d]$ has $n - k$ rows and n columns. The product xH^T is a vector of length $n - k$. This is also known as the **syndrome** of x. We have seen that a vector $x \in \mathbb{F}_q^n$ is an element of $\mathcal{C}$ if and only if its syndrome is the 0-vector. We have described the Hamming and Simplex codes by their generator and check matrices.

Another useful method of describing a linear code is geometrical in nature. It is useful to introduce a geometrical structure, which is of fundamental importance anyway.

Definition (8.1) (projective geometry) *Let V be an $(n+1)-$dimensional vectorspace over $\mathbb{F}_q$. Call the $(i+1)$-dimensional subspaces of V the i-**flats** of our geometry. Here $i = 1, 2, \ldots, n$. Incidence is defined by inclusion. That is to say flats a and b are called* **incident** *if either $a \subset b$ or $b \subset a$. These flats and their incidence relation define the n-dimensional* **projective geometry** *$\mathcal{P}_n(q)$ of* **order** *q.*

- *0-flats are* **points** *(=1-dimensional subspaces),*
- *1-flats are* **lines** *(=2-dimensional subspaces),*
- $\ldots$
- *$(n-1)-$flats are* **hyperplanes***(=n-dimensional subspaces).*

Observe the shift of 1 in the notions of dimension: a vectorspace of dimension $n+1$ gives rise to the projective geometry of dimension n. The **projective line** $\mathcal{P}_1(q)$ is just a collection of $q+1$ points (the 1-dimensional subspaces of a 2-dimensional vectorspace). This is therefore geometrically trivial. As a first nontrivial example consider $\mathcal{P}_2(2)$, the projective plane of order 2. It has 7 points and 7 lines. Each line has 3 points, each point is on 3 lines. The reader is urged to check that this leads to the following picture. $\mathcal{P}_2(2)$ is also known as the **Fano plane.** See Figure **??**.

The following is an easy exercise. For the sake of completeness we include a proof:

Definition (8.2) (Gauss polynomial) *The number of i-flats of $\mathcal{P}_n(q)$ is*

$$\begin{bmatrix} n+1 \\ i+1 \end{bmatrix} = \frac{(q^{n+1}-1)(q^{n+1}-q)\cdot\ldots(q^{n+1}-q^i)}{(q^{i+1}-1)(q^{i+1}-q)\cdot\ldots(q^{i+1}-q^i)} \tag{8.2}$$

In particular the number of points and the number of hyperplanes is $(q^{n+1}-1)/(q-1)$, which is clear anyway.

Proof: The numerator of the expression above is the number of ordered $(i+1)$-tuples of linearly independent elements. Each such set generates an i-flat. The multiplicity with which a given i-flat is counted is the number of

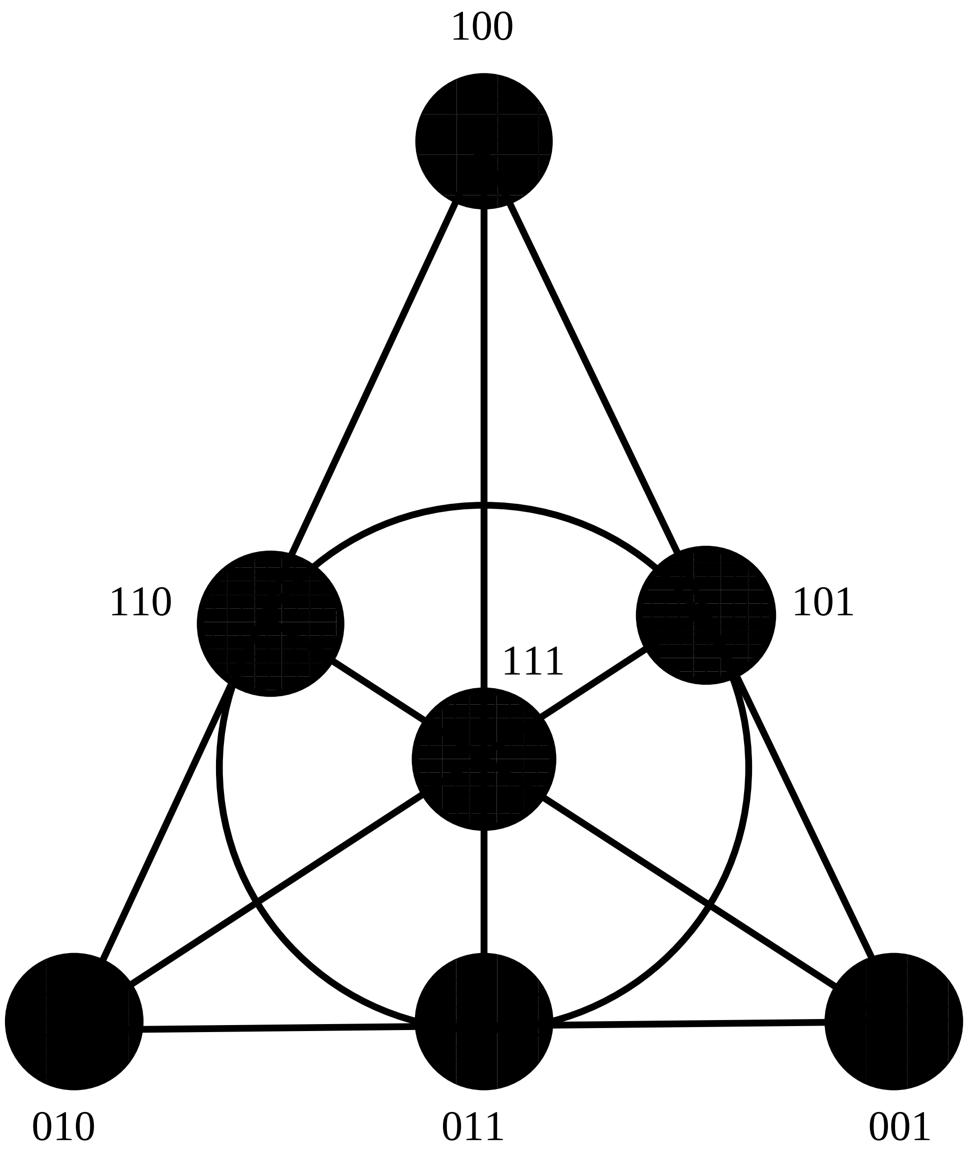

Figure 8.1: **The Fano plane**

its ordered bases. This is the denominator of the expression above. ■

Our geometrical method of representing a linear code is the following: consider the columns of a generator matrix as generators of points in $\mathcal{P}_{k-1}(q)$. Then our code is identified with a set of points in projective $(k-1)$-dimensional geometry. In fact, one should be a little more careful. It is perfectly possible that different columns of G are multiples of each other or even equal as k-tuples. In view of this one should describe the code as a multiset of projective points, so that each projective point may contribute $0, 1, 2, \dots$ columns. According to standard terminology a code is called **projective** if no projective point occurs with multiplicity > 1. Let us call the description of a code by means of the projective points generated by the columns of a generator matrix the **geometrical description** of a code. The geometrical description of the Simplex codes is particularly easy: $\mathcal{S}_k(q)$ is described as the set of all points of $\mathcal{P}_{k-1}(q)$. The Simplex code $\mathcal{S}_3(2)$ as given in the beginning of Chapter 7 is geometrically equivalent to the Fano plane.

Why does it suffice to consider the 1-dimensional subspaces instead of the columns themselves? This leads to the following definition:

Definition (8.3) *Two linear q-ary codes $\mathcal{C}_1, \mathcal{C}_2$ are* **diagonally equivalent** *if there is a vector $(\lambda_1, \dots, \lambda_n)$ with nonzero entries such that the mapping $(x_1, x_2, \dots, x_n) \longrightarrow (\lambda_1 x_1, \dots, \lambda_n x_n)$ is a bijection between $\mathcal{C}_1$ and $\mathcal{C}_2$.*

Diagonal equivalence may obviously be described by the action of diagonal matrices. It is clear that diagonally equivalent codes have essentially the same structure. In particular they agree in length, dimension and minimum distance. Replacing columns of G by a nonzero scalar multiples amounts to replacing the code by a diagonally equivalent code. The geometrical description gives us an equivalence class of diagonally equivalent codes.
A variant of this method is what we wish to call the **dual geometrical description:** describe the code as a multiset of projective points in $\mathcal{P}_{n-k-1}(q)$ generated by columns of a check matrix. This means that we use the geometrical description for the dual code.
If our code is described by the geometrical method, how are the basic parameters minimum distance and strength determined? Let $G = (a_{ij}), i = 1, 2, \dots, k; j = 1, 2, \dots, n$ be a generator matrix of $\mathcal{C}$. This gives us a basis $v_i, i = 1, 2, \dots, k$ of the code, where v_i is row i of G. We interpret column j of

G as a point in $\mathcal{P}_{k-1}(q)$. An arbitrary code word has the form $v = \sum_{i=1}^{k} \alpha_i v_i$. Its entry in column j is $\sum_{i=1}^{k} \alpha_i a_{ij} = (\alpha_1, \alpha_2, \ldots, \alpha_k) \cdot P_j$. The entry is zero if and only if P_j is contained in the hyperplane (=($k-1$)-dimensional vectorspace) $(\alpha_1, \alpha_2, \ldots, \alpha_k)^{\perp}$.

Proposition (8.1) *Let the q-ary linear code $\mathcal{C}$ be described by the geometrical method. This means that we have a generator matrix $G = (a_{ij}), i = 1, 2, \ldots, k; j = 1, 2, \ldots, n$, whose rows v_i form a basis of $\mathcal{C}$, and we interpret the columns P_j of G as describing points in $\mathcal{P}_{k-1}(q)$. Let $v = \sum_{i=1}^{k} \alpha_i v_i \neq 0$ be an arbitrary nonzero element of $\mathcal{C}$. Then we have*

$$wt(v) = n- \mid (\alpha_1, \alpha_2, \ldots, \alpha_k)^{\perp} \cap \{P_1, \ldots, P_n\} \mid .$$

So if $\mathcal{P} = \{P_1, \ldots, P_n\}$ denotes the set of projective points describing the columns, then the weights of $\mathcal{C}$ are the numbers $n- \mid E \cap \mathcal{P} \mid$, where E varies over the hyperplanes of $\mathcal{C}$.
We use a toy example to illustrate: what is the maximum d of a binary code $[11, 3, d]$? The Griesmer bound Theorem (4.4) shows that $d \leq 6$. (a 3-dimensional code of minimum distance 7 has length ≥ 13 by the Griesmer bound). Does a code $[11, 3, 6]_2$ exist? By the geometrical description such a code can be described by a weight function w defined on the points of the Fano plane, with natural numbers as values. Put $w(l) = \sum_{P \in l} w(P)$ for every line l. Then the weight function w describes a code $[11, 3, 6]_2$ if and only if the following are satisfied:

1. The sum of the weights over all points P of the Fano plane is 11, and
2. $w(l) \leq 11 - 6 = 5$ for every line

Such a function w is obtained by choosing a quadrangle $\mathcal{Q} \subset \mathcal{P}_2(2)$ and putting $w(P) = 2$ if $P \in \mathcal{Q}$, $w(P) = 1$ otherwise.

The corresponding generator matrix for our code $[11, 3, 6]_2$ is as follows:

1	1	0	0	1	1	1	1	1	1	0
0	0	1	1	0	0	1	1	0	1	1
0	0	0	0	1	1	1	1	1	0	1

How about the strength of a code $\mathcal{C}$ described by the geometrical method? This is related to the following concept:

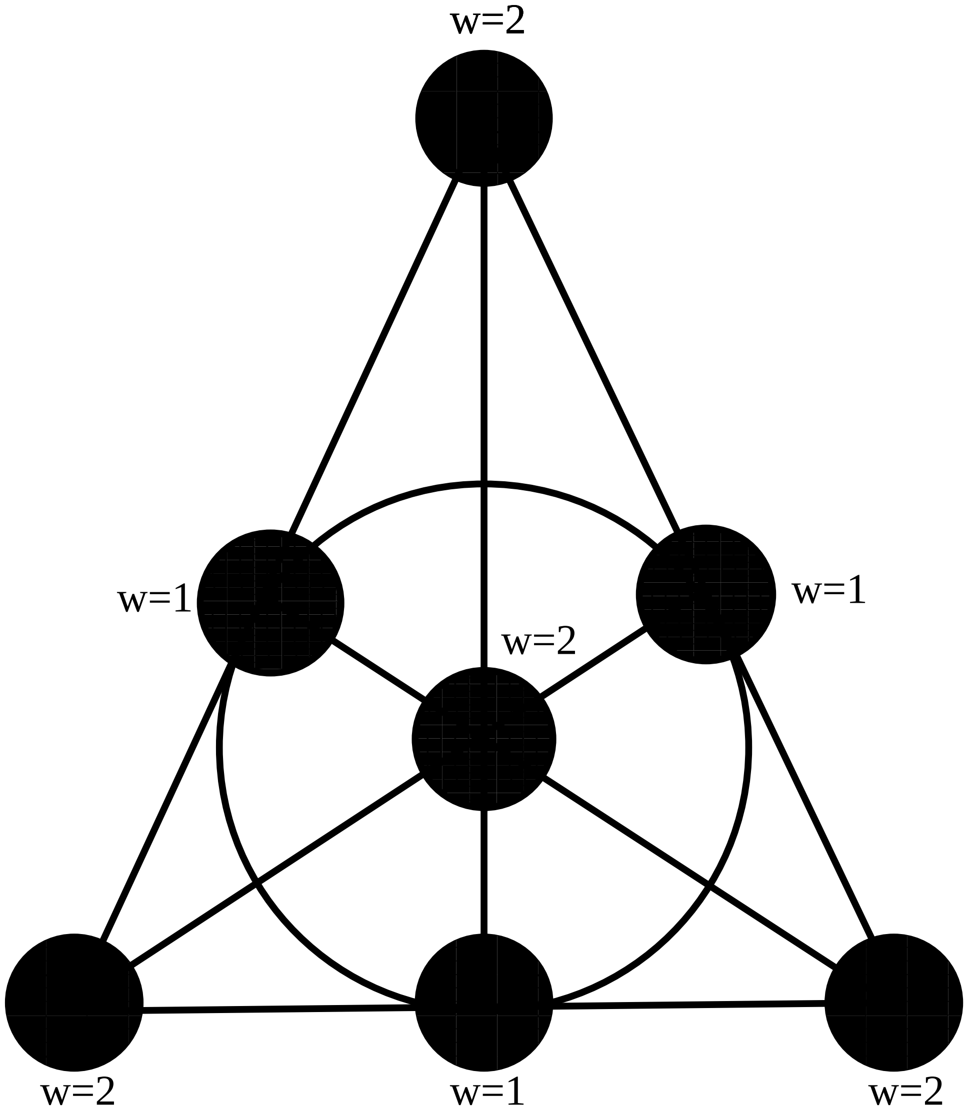

Figure 8.2: **A binary code [11,3,6]**

Definition (8.4) *Points* $P_1, P_2, \ldots, P_n$ *of* $\mathcal{P}_{k-1}(q)$ *are in* **general position** *if they generate a subspace* $\mathcal{P}_{n-1}(q)$. *Equivalently this means that the corresponding 1-dimensional subspaces generate an* n*-dimensional vectorspace, so are linearly independent. Another way of expressing this is the following: consider the matrix with* k *rows and* n *columns, whose columns represent the* P_j. *Then the* P_j *are in general position if and only this matrix has rank* n.

Proposition (8.2) *Let the* q*-ary linear code* $\mathcal{C}$ *be described by the geometric method like in Proposition (8.1). The strength of* $\mathcal{C}$ *is the maximum number* t *such that any set of* t *columns (projective points) is in general position.*

Let us record what this means for small values of t : the strength is > 1 if and only if the P_j are pairwise different. We have called such codes **projective codes** earlier. The strength is ≥ 3 if and only if no three of the P_j are on a line. The strength is ≥ 4 if and only if no four of the P_j are contained in a plane.
The **trace-method** is closely related to the geometrical method: Let $F = \mathbb{F}_q^n$ and impose the structure of a field $\mathbb{F}_{q^n}$ on F. Pick nonzero elements $x_i \in F, i = 1, 2, \ldots, n$. Define a q-ary code of length n whose codewords are indexed by the $v \in F$. The entry in coordinate i is defined as $tr(vx_i)$. In order to make the comparison with the geometrical method choose a basis $v_1, v_2 \ldots, v_k$ of F and write $x_i = \sum_{j=1}^k \alpha_{ji} v_j$. Write a generator matrix with respect to the dual basis $w_1, w_2, \ldots, w_k$ (see Theorem (2.6)). The entry in row j and column i of this generator matrix is $tr(w_j x_i) = \alpha_{ji}$. We see that the code described by the trace-method is the same as the code given by the geometrical method, corresponding to the points $P_i = < x_i >, i = 1, 2, \ldots, n$
Our final method of describing a code is the **linear functional description:** Fix a k-dimensional vector space V. Describe a code as a family of n linear functionals $\phi_i : V \longrightarrow \mathbb{F}_q, i = 1, 2, \ldots, n$. Each linear functional describes a coordinate of the code. The direct description is obtained from this by writing out the matrix whose rows are indexed by the elements of $x \in V$, whose columns are indexed by the ϕ_i, with corresponding entry $\phi_i(x)$.
Each of these description methods has its advantages in certain situations. For example, construction X (Theorem (10.2)) is most naturally described using the linear functional description.

8.1 Some codes related to conic sections

Let $p(X_1, X_2, \ldots, X_k)$ be a homogeneous polynomial in k variables, with coefficients in $I\!F_q$. If $(x_1, x_2, \ldots, x_k)$ is a root of p, then all scalar multiples $\lambda(x_1, x_2, \ldots, x_k), \lambda \in I\!F_q$ are roots as well. We can therefore consider the roots of the homogeneous polynomial as a set of points in projective space $\mathcal{P}_{k-1}(q)$. This set is denoted by $V(p)$ (the **variety** of $p(X_1, x_2, \ldots, X_k)$.) If the degree of p is 2 we call $V(p)$ a **quadric.** For calculations in $\mathcal{P}_{k-1}(q)$ we use **homogeneous coordinates:** $(x_1 : x_2 : \ldots : x_k)$ denotes the point (=1-dimensional subspace) generated by $(x_1, x_2, \ldots, x_k)$.

Theorem (8.1) *The quadric* $\mathcal{Q} = V(Y^2 - XZ) \subset \mathcal{P}_2(q)$ *has* $q+1$ *points, more precisely* $\mathcal{Q} = \{(0:0:1)\} \cup \{(1:y:y^2) \mid y \in I\!F_q\}$. *No three points of* $\mathcal{Q}$ *are on a line.*

Only the last claim of the Theorem deserves a proof. Because of Definition (8.4) and Proposition (8.2) from Chapter 8 this follows from the fact that matrices $\begin{pmatrix} 1 & a^2 & b^2 \\ 0 & a & b \\ 0 & 1 & 1 \end{pmatrix}$ and $\begin{pmatrix} a^2 & b^2 & c^2 \\ a & b & c \\ 1 & 1 & 1 \end{pmatrix}$ are nonsingular.$\blacksquare$

We sketch how to obtain some interesting 3-dimensional codes related to the conic section as described in Theorem (8.1). Some computational details are left to the reader as an exercise.

Let q be an odd prime power. Prove the following facts:

- There are $\binom{q+1}{2}$ **secants** (lines containing 2 points of $\mathcal{Q}$) and $q+1$ tangents (lines containing one point of $\mathcal{Q}$). Each point of $\mathcal{Q}$ is on one tangent.
- Each point not in $\mathcal{Q}$ either is on precisely 2 tangents (these are the **outer points**) or on no tangent at all (these are the **inner points**).
- There are $\binom{q+1}{2}$ outer points and $\binom{q}{2}$ inner points.
- Each secant has $(q-1)/2$ inner points ans equally many outer points.
- Each tangent has q outer points.

- Each line avoiding $\mathcal{Q}$ has $(q+1)/2$ inner points and equally many outer points.

Use the geometric description to construct codes in the following way:

Using the inner points and one point of the quadric show that we obtain a code

$$[\binom{q}{2}+1, 3, (q-1)^2/2]_q \ (q \text{ odd})$$

Using the inner points and all points of $\mathcal{Q}$ we obtain codes

$$[\binom{q+1}{2}+1, 3, (q^2-1)/2]_q \ (q \text{ odd})$$

1. Some methods of defining linear codes are:
2. The **direct method** (used for the Reed-Solomon codes).
3. **Generator matrix** or **check matrix** (used for the Hamming and Simplex codes)
4. The **geometric method** and the **dual geometric method.**
5. The **trace method**.
6. The **linear functional method.**

1. The general notion underlying the geometric method is that of **projective spaces** $PG_n(q)$ (**dimension** n and **order** q).
2. Let $V = V(n+1, q)$ be a vector space of dimension $n+1$ over $\mathbb{F}_q$.
3. The **points** of $PG_n(q)$ are the 1-dimensional subspaces of V.
4. The **lines** of $PG_n(q)$ are the 2-dimensional subspaces of V.
5. The **hyperplanes** of $PG_n(q)$ are the n-dimensional subspaces of V.

1. The **geometric method:** view the columns of a generator matrix G as a family of points from $PG_{k-1}(q)$.
2. Write $w(P)$ if point P occurs $w(P)$ times as a column of G.
3. w is a function defined on the points of $PG_{k-1}(q)$ with non-negative integer values.
4. The length is $n = mass(w) = \sum_P w(P)$.
5. For every hyperplane H let $w(H) = \sum_{P \in H} w(H)$.
6. The minimum distance is $d = n - max_H w(H)$.

8.2 Problems

1. Assume you have a set of 17 points in $\mathcal{P}_3(4)$ with the property that every hyperplane contains at most 5 of these points. What are the parameters of the corresponding quaternary code?
2. What does the Griesmer bound tell us about the maximum possible minimum distance of a binary code $[14, 3, d]_2$? Construct such a code by the geometric method.

Chapter 9

Denniston arcs and Denniston codes

We have seen how to describe linear codes by sets of points in projective geometry. It is not surprising that point sets with extremal properties lead to particularly interesting codes. This is the case of the Denniston arcs, which we want to construct in this section:
we work in the projective plane $PG(2,q)$. Remember that the points and lines are simply the 1-dimensional respectively the 2-dimensional subspaces of a vector space of dimension three. This makes it easy to check that the following are satisfied:

- $PG(2,q)$ has q^2+q+1 points and equally many lines.
- Each line has $q+1$ points, each point is on $q+1$ lines.
- Each pair of different points is on precisely one line, each pair of different lines has precisely one point in common.

Definition (9.1) *A set of points $K \subseteq PG(2,q)$ is a k-**arc** if no line contains more than k points of K.*

We fix k and ask what the maximum cardinality of k-arc in $PG(2,q)$ may be. Fixing a point $P \in K$ and counting the points of K on the lines through P we clearly obtain an upper bound:

$$| K | \leq 1 + (q+1)(k-1) = qk + k - q \tag{9.1}$$

Definition (9.2) *A k-arc $K \subseteq PG(2,q)$ of cardinality $qk+k-q$ is a* **maximal k-arc.**

Clearly $1 \leq k \leq q+1$. A point is a maximal 1-arc, the set of all points is a maximal $(q+1)$-arc, the set of points off a certain line is a maximal q-arc. These are the trivial cases. We ask for which pairs (q,k) maximal k-arcs exist in $PG(2,q)$, where $2 \leq k < q$. Our counting argument shows that every line must intersect K either in the empty set or in cardinality k. Fixing a point $Q \notin K$ (such a point exists if $k < q+1$) we see that $\mid K \mid$ is a multiple of k. It follows that $k|q$.

Proposition (9.1) *A maximal k-arc in $PG(2,q), k \leq q$ can exist only if k divides q.*

We are led to the following question: let $q = p^a, p$ prime, $a > 1$. For which i, where $0 < i < a$ does there exist a maximal p^i-arc in $PG(2,q)$.? We will describe an elegant construction due to Denniston, which shows that maximal arcs always exist when $p = 2$. We find it handy to work in the affine plane.

Definition (9.3) *Fix a line g in $PG(2,q)$. The incidence structure induced on the q^2 points off g by the q^2+q lines $\neq g$ is an* **affine plane.**

Let us introduce coordinates: start from the 3-dimensional space $\mathbb{F}_q^3$. Write $(a:b:c)$ for the point (=1-dimensional subspace) generated by the triple $(a,b,c) \neq (0,0,0)$. Clearly $(a:b:c) = (\lambda a : \lambda b : \lambda c)$ for every $0 \neq \lambda \in \mathbb{F}_q$. Fix a line g and call it the **line at infinity.** We choose $g = <(1:0:0),(0:1:0)>$ as line at infinity. The points off g (the points of our affine plane) are called **affine points.** The affine points are precisely those with nonzero last coordinate. Each such point can be written uniquely as $(x:y:1)$. We will write $P(x,y) = (x:y:1)$. Consider the lines through points $\in g$ (**points at infinity**). The $q+1$ points at infinity are $(0:1:0)$ and $(1:m:0), m \in \mathbb{F}_q$. The line through $(0:1:0)$ and $(c:y:1)$ passes through point $(0:1:0)$ at infinity. Its affine points are the points $(c:y:1), y \in \mathbb{F}_q$. This line has equation $x = c$. The line through $(1:0:0)$ and $(x:c:1)$ has equation $y = c$. For $m \neq 0$ the line through $(1:m:0)$ and $(a:b:1)$ consists of the points $(\lambda + a : \lambda m + b : 1)$. This line has equation

$y = mx + c$, where $c = b - am$. Naturally we call m the **slope** of the line $y = mx + c$. The lines $x = c$ have slope ∞.

We can define the Denniston arcs:

Definition (9.4) (Denniston arcs) *Let $q = 2^a, 0 < i < a$. Denote by $tr = tr : \mathbb{F}_q \longrightarrow \mathbb{F}_2$ the* **trace**, *choose an element $b \in \mathbb{F}_q$ such that $tr(b) = 1$, put $Q(X,Y) = X^2 + XY + bY^2$. Let $E \subset \mathbb{F}_q$ be an i-dimensional $\mathbb{F}_2$-vectorspace. The corresponding Denniston arc $\mathcal{D}$ is a subset of the affine plane. It is defined as*

$$\mathcal{D} = \{P(x,y) \mid Q(x,y) \in E\} \tag{9.2}$$

We have to prepare the ground by considering quadratic functions.

Lemma (9.1) *Let $q = 2^a, F = \mathbb{F}_q$. The image $f(F)$ of a quadratic function $f(X) = aX^2 + bX + c$, where $a \neq 0$, is either a hyperplane H (=$\mathbb{F}_2$-subspace of dimension $a-1$) or the complement $F - H$ of a hyperplane. Moreover every image of f has precisely 2 preimages.*

Proof: Consider $g(X) = aX^2 + bX$. As $x \longrightarrow x^2$ is a field automorphism we see that $g(X)$ yields an $\mathbb{F}_2$-linear mapping. Its kernel consists of 0 and b/a, so has dimension 1. Its image must be a hyperplane H. If $c \in H$, then $f(F) = H$. If $c \notin H$, then $f(F) = F - H$.■

Theorem (9.1) *The point set $\mathcal{D}$ of Definition (9.4) is a maximal 2^i-arc. Thus for every $i \leq a$ there is a maximal 2^i-arc in $PG(2, 2^a)$.*

Proof: We have defined $\mathcal{D}$ as a point set in the affine plane $AG(2,q)$. As $AG(2,q) \subset PG(2,q)$ we may consider it as a subset of the projective plane. We remark at first that it suffices to show that every line intersects $\mathcal{D}$ either in 0 or in 2^i points (fix a point $P \in \mathcal{D}$, count the points of $\mathcal{D}$ on the lines through P).
$Q(X,Y)$ is by definition a quadratic form, in particular
$Q(\lambda X, \lambda Y) = \lambda^2 Q(X,Y)$. For which pairs $(x,y) \in F \times F$ if $Q(x,y) = 0$? If $y = 0$, then $Q(x,0) = x^2 = 0$ implies $x = 0$. If $y \neq 0$, then without restriction $y = 1$, hence $Q(x,1) = x^2 + x + b$. Assume $Q(x,1) = 0$. As squaring is a field automorphism we have $tr(x) = tr(x^2)$, hence $0 = tr(0) = tr(b)$, contradiction

as b was chosen such that $tr(b) = 1$. We conclude that $Q(x, y) = 0$ if and only if $x = y = 0$. Such quadratic forms are called **anisotropic.**
Consider at first lines through the origin: In the case of the line $x = 0$ we get $Q(0, y) = by^2$, so this line intersects $\mathcal{D}$ in 2^i points. In the case of the lines $y = mx$ we get $Q(x, mx) = x^2 + mx^2 + bm^2x^2 = \frac{1}{b}Q(bm, 1)x^2$. As $Q(bm, 1) \neq 0$ we see that with $x \in F$ this varies over all elements in F. We get an intersection size of 2^i again.
So consider lines not through the origin. If (x, y) varies through the q affine points of such a line l, then $Q(x, y)$ is a quadratic expression. As $P(0, 0)$ is not on l, we conclude from Lemma (9.1) that $Q(l)$ is the complement of a hyperplane H, where every image occurs exactly twice. To finish the proof we have to consider two cases:
if $E \subseteq H$, then clearly l contains no point of $\mathcal{D}$. In the contrary case we have $\mid E \cap H \mid = 2^{i-1}$. As every image occurs twice we conclude that l intersects $\mathcal{D}$ in cardinality 2^i.■

The construction makes heavy use of the fact that we are in characteristic 2. Every hope of generalizing this construction to odd characteristic has been shattered by a recent result of Ball, Blokhuis and Mazzocca [3], where it is shown, that nontrivial maximal k-arcs never exist in odd characteristic.
In order to construct codes out of Denniston arcs we use the geometric description. For each $P \in \mathcal{D}$ choose a generator y_P of P (thus $P = \mathbb{F}_q y_P$). A generator matrix of the code has the $y_P, P \in \mathcal{D}$ as columns. What are the parameters of this linear q-ary 3-dimensional code $\mathcal{C}$? Its length n equals the cardinality of $\mathcal{D}$, thus $n = 2^{a+i} + 2^i - 2^a$. Let $x \in \mathcal{C}$ be a nonzero codeword. Its weight is by definition $wt(x) = n - \mid x^\perp \cap \mathcal{D} \mid$. As $x^\perp$ is a 2-dimensional space, hence a line of $PG(2, q)$, there are only two possibilities: either $\mid x^\perp \cap \mathcal{D} \mid = 0$ or $\mid x^\perp \cap \mathcal{D} \mid = 2^i$. It follows that either $wt(x) = n$ or $wt(x) = n - 2^i$.

Definition (9.5) *Let $q = 2^a, i < a$ and let $\mathcal{D}$ be the Denniston 2^i-arc of Definition (9.4). We define the* **Denniston code** *as a 3-dimensional q-ary code of length $n = \mid \mathcal{D} \mid = 2^{a+i} + 2^i - 2^a$ whose generator matrix has the $y_P, P \in \mathcal{D}$ as columns.*

Theorem (9.2) *The Denniston code from Definition (9.5) is a linear q-ary code with parameters $[2^{a+i} + 2^i - 2^a, 3, 2^{a+i} - 2^a]$. Moreover every nonzero codeword has one of two weights, either $w_1 = d = 2^{a+i} - 2^a$ or $w_2 = n = 2^{a+i} + 2^i - 2^a$.*

Observe that the Denniston codes meet the Griesmer bound (Theorem (4.4)) with equality. Codes satisfying the property that there are only two nonzero weights are known as **two-weight codes.** They are particularly interesting because of their relationships with structures such as **strongly regular graphs** and **partial geometries.** Our geometrical description makes it particularly easy to determine the number of code vectors of weights w_1 and w_2, respectively (see the Problems section). In the special case $i = 1$ the Denniston arcs ($q + 2$ points, no three on a line) are known as **hyperovals.** In the following subsections we describe the hyperovalcodes over $\mathbb{F}_4$ and $\mathbb{F}_8$.

9.1 The hexacode

Consider the case $a = 2, i = 1$ of the Denniston arcs. This is a hyperoval in the projective plane $PG(2, 4)$ of order 4. As $tr(\omega) = 1$ we can take the quadratic form Q to be $Q(X, Y) = X^2 + XY + \omega Y^2$. The 1-dimensional subspace of $\mathbb{F}_4$ will be chosen to be $\mathbb{F}_2$. The hyperoval is then

$$\mathcal{D} = \mathcal{O} = \{P(0, 0), P(0, \omega), P(1, 0), P(1, \omega^2), P(\omega, \omega), P(\omega, \omega^2)\}.$$

A generator matrix of the corresponding hyperovalcode is obtained by by using as columns the coordinates of the points $P \in \mathcal{O}$. It follows that the following is a generator matrix of this code:

0	0	1	1	ω	ω
0	ω	0	ω^2	ω	ω^2
1	1	1	1	1	1

This quaternary code $[6, 3, 4]_4$ is also known as the **hexacode.** Our geometrical description makes it easy to calculate the weight distribution of the Denniston codes (see the Problems section). The hexacode has 45 words of weight 4 and 18 words of weight 6. In the next section we will use the hexacode to give a construction of the single most famous code, the binary Golay code.

9.2 A code over $\mathbb{F}_8$.

In case $i = 1, a = 3$ we can choose $b = 1$ as $tr(1) = 1$. The quadratic form is therefore $Q(X,Y) = X^2 + XY + Y^2$. As before choose $E = \mathbb{F}_2$. Observe in particular that Q is symmetric in X and Y, so the same is true of the points $P(x,y) \in \mathcal{O}$. We give the generator matrix whose columns are the points of $\mathcal{O}$:

0	0	1	1	ϵ^3	ϵ^5	ϵ^3	ϵ^6	ϵ^5	ϵ^6
0	1	0	1	ϵ^5	ϵ^3	ϵ^6	ϵ^3	ϵ^6	ϵ^5
1	1	1	1	1	1	1	1	1	1

The code is linear 8-ary, with parameters $[10,3,8]_8$. The only nonzero weights occuring are 8 and 10.

1. Let $q = 2^a, i < a$. The corresponding **Denniston code** is q-ary of dimension 3, with length $n = 2^{a+i} + 2^i - 2^a$. Every nonzero codeword has one of the two weights $d = 2^{a+i} - 2^a$ or $2^{a+i} + 2^i - 2^a$.

2. The Denniston codes meet the Griesmer bound with equality.

3. Geometrically the Denniston code is described as a set of points in the projective plane meeting every line in either 0 or 2^i points.

4. The **hexacode** $[6,3,4]_4$ and code $[10,3,8]_8$ have been described explicitly.

9.3 Problems

1. Determine the weight polynomial of the 8-ary Denniston code as given in Section 9.2.

2. Determine the numbers A_{w_1} and A_{w_2} of words of weight w_1 and w_2, respectively, in the Denniston codes.

Chapter 10

Some basic recursive constructions

10.1 Shortening and projection

Let $\mathcal{C}$ be a linear subspace of $\mathbb{F}_q^n$. We consider at first the following rather trivial operations:

Definition (10.1) (shortening) *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a k-dimensional subspace. Fix a coordinate, say n. Define the* **shortening** *$\mathcal{D}$ by*

$$\mathcal{D} = \{(x_1, x_2, \ldots, x_{n-1}) \mid (x_1, x_2, \ldots, x_{n-1}, 0) \in \mathcal{C}\}.$$

Then $\mathcal{D}$ has dimension $\geq k-1$. If $\mathcal{C}$ has minimum distance d, then $\mathcal{D}$ has minimum distance $\geq d$.

In fact, $\mathcal{D}$ is a linear subspace of $\mathcal{C}$. So its minimum distance cannot be smaller than that of $\mathcal{C}$. As $\mathcal{D}$ is by definition the kernel of a linear functional $: \mathcal{C} \longrightarrow \mathbb{F}_q$, its codimension in $\mathcal{C}$ cannot exceed 1. Observe that $\mathcal{D} = \mathcal{C}$ if and only if the codewords of $\mathcal{C}$ vanish in the last coordinate. If $\mathcal{C}$ has parameters $[n, k, d]_q$, where $k > 0$, then the shortened code has parameters $[n-1, k-1, d]_q$.

Definition (10.2) (projection) *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a k-dimensional subspace. Fix a coordinate, say n. Define the* **projection** *$\mathcal{D}$ by*

$$\mathcal{D} = \{(x_1, x_2, \ldots, x_{n-1}) \mid \text{ there is } x \in \mathbb{F}_q \text{ such that } (x_1, x_2, \ldots, x_{n-1}, x) \in \mathcal{C}\}.$$

If $d(\mathcal{C}) > 1$, then $dim(\mathcal{C}) = dim(\mathcal{D})$. If $\mathcal{C}$ has minimum distance d, then $\mathcal{D}$ has minimum distance $\geq d-1$.

It is in fact clear that the minimum distance of $\mathcal{D}$ can differ by at most 1 from that of $\mathcal{C}$. If the dimension were smaller, then the projection $\phi : \mathcal{C} \longrightarrow \mathcal{D}$ given by $\phi(x_1, \ldots, x_{n-1}, x_n) = (x_1, \ldots, x_{n-1})$ would have a nontrivial kernel, which would imply $d = 1$. The statement concerning the strength is obvious. In particular every code $[n, k, d]$ allows the construction, in a trivial way, of codes $[n-1, k-1, d]$ (a subcode) and $[n-1, k, d-1]$ (a projection).
The following is a slight generalization of the process that led us to the Griesmer bound:

Theorem (10.1) (residual code) *Let $\mathcal{C}$ be a q-ary linear code with parameters $[n, k, d]$ and $v \in \mathcal{C}$ a word of weight w. Assume $d - w + \lceil w/q \rceil > 0$. Then there is a code $[n-w, k-1, d-w+\lceil w/q \rceil]$ (the* **residual code***) obtained by projecting a subcode of $\mathcal{C}$ to the coordinates which do not belong to the support of v.*

Proof: Proceed as in the proof of (4.4), based on the word $v = v_1$ of weight w. Complete v_1 to a basis $\{v_1, \ldots, v_k\}$, let $\mathcal{D}$ be the code obtained by projecting $< v_2, \ldots, v_k >$ to the coordinates which do not belong to the support of v. Let x be a nonzero word of $\mathcal{D}$ having weight w' after projection to $\mathcal{D}$. We want a lower bound on w'. It can be assumed that the nonzero entries of v are 1. Denote by δ_i the number of coordinates in the support of v, where x has entry $i \in \mathbb{F}_q$. As $x - i \cdot v$ is a nonzero word of $\mathcal{C}$ we get $wt(x - i \cdot v) = w' + w - \delta_i \geq d$, or $w' \geq d - w + \delta_i$. As the mean value of the δ_i is w/q, our claim concerning the minimum distance of the residual code is proved. The same considerations also show that the residual code does indeed have dimension $k-1$. ■

10.2 The binary case

Let $x = (x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$. We can read x as the characteristic function of a subset of $\{1, 2, \ldots, n\}$. In that perspective binary codes are families of subsets of a fixed set of cardinality n. Here n is the length. This natural connection with extremal problems on families of sets is a peculiarity of the binary case. As an example, let $n = 8$. Then $x = (0,0,1,1,0,0,0,1) \longrightarrow \{3,4,8\}$. Likewise $(0,0,0,0,0,0,0,0) \longrightarrow \emptyset$. We set that $wt(x)$ is the cardinality of the corresponding set.

Lemma (10.1) *Let v, w be binary codewords of length n. Define $v \cap w$ to be the binary word with entries 1 at all coordinates where both v and w have entry 1. Then $wt(v + w) = wt(v) + wt(w) - 2wt(v \cap w)$.*

This is trivial. The definition of $v \cap w$ becomes natural if one identifies v, w with sets as above.

Lemma (10.2) *A binary linear code $\mathcal{C}$ all of whose weights are multiples of 4 is contained in its dual code: $\mathcal{C} \subseteq \mathcal{C}^{\perp}$.*

Proof: Let $v, w \in \mathcal{C}$. By the preceding lemma we have $wt(v + w) = wt(v) + wt(w) - 2wt(v \cap w)$. As $\mathcal{C}$ is linear we have $v + w \in \mathcal{C}$. By assumption each of $wt(v), wt(w)$ and $wt(v + w)$ is divisible by 4. It follows that $wt(v \cap w)$ is even. As $v \cdot w$ is the parity of $v \cap w$ we conclude $v \cdot w = 0$.■

Proposition (10.1) (parity check bit) *Let $\mathcal{C}$ be a binary linear code $[n, k, d]_2$, where d is odd. We define a code $\tilde{\mathcal{C}}$ in the following way:*
If $x = (x_1, x_2, \ldots, x_n) \in \mathcal{C}$, then $\tilde{x} = (x_1, x_2, \ldots, x_n, u) \in \mathcal{C}$, where u is chosen such that $wt(\tilde{x})$ is even (equivalently $\tilde{x} \cdot \tilde{x} = 0$). Then u is called a parity check bit. $\tilde{\mathcal{C}}$ has parameters $[n + 1, k, d + 1]$.

In the next section we will see this construction as a special case of a more general method.

10.3 Lengthening

We describe a simple process, which yields a lengthening of a given code with larger minimum distance provided we know a subcode with a larger minimum distance. This has been described as **construction X** in [15].

Theorem (10.2) (construction X) *Let $\mathcal{C} \supset \mathcal{D}$ be q-ary linear codes, with parameters $[n, k, d] \supset [n, k - \kappa, d']$, where $d' > d$. Assume further there is a code $\mathcal{E}$, which we will call the auxiliary code, with parameters $[l, \kappa, \delta]$. Then there is a l-step lengthening of $\mathcal{C}$ with parameters $[n + l, k, min(d + \delta, d')]$.*

Proof: We describe the additional l columns as linear functionals $\phi : \mathcal{C} \longrightarrow \mathbb{F}_q$. More precisely we use only linear functionals having the subspace $\mathcal{D}$ in the kernel. We can therefore write $\phi : \mathcal{C}/\mathcal{D} \longrightarrow \mathbb{F}_q$. Identify $\mathcal{C}/\mathcal{D}$ with $\mathcal{E}$ and use the linear functionals describing the code $\mathcal{E}$. Each element of $\mathcal{D}$ obtains entries 0 in the new coordinates. Each word from $\mathcal{C} - \mathcal{D}$ gets $\geq \delta$ nonzero entries in the new coordinates. This proves the claim.■

The easiest and best-known application of construction X is the addition of a parity-check bit in the binary case (see the preceding section). In fact, the subset $\mathcal{D}$ of words of even weight of $\mathcal{C}$ is a linear subcode of codimension 1 (see the Problems section). Apply (10.2) with the trivial code $[1, 1, 1]$ as auxiliary code. This is another description of the parity check.

Corollary (10.1) (parity check) *Let $\mathcal{C}$ be a binary code $[n, k, d]$, where d is odd. Application of construction X (10.2) with the even-weight subcode $\mathcal{D} \subset \mathcal{C}$ and $[1, 1, 1]$ as auxiliary code yields a code with parameters $[n+1, k, d+1]$.*

We have seen the binary Hamming code $\mathcal{H}_2(3)$ in Definition (7.2). A check matrix has been given shortly before Definition (7.2). If we use equation 7.2 we can determine the weight polynomial of $\mathcal{H}_2(3)$. It is $X^7+7X^4Y^3+7X^3Y^4+Y^7$. When we add a parity check bit to this code we obtain a code, which we will denote by $\mathcal{H}_8$. It is called the **extended Hamming code** $[8, 4, 4]_2$. As each word of odd length of $\mathcal{H}_2(3)$ is lengthened by 1, the weight polynomial of $\mathcal{H}_8$ is $X^8 + 14X^4Y^4 + Y^8$. We conclude from Theorem (10.2) that $\mathcal{H}_8$ is selfdual.

Definition (10.3) *The binary extended Hamming code $\mathcal{H}_8$ with parameters $[8, 4, 4]$ is obtained by adding a parity check to the Hamming code $\mathcal{H}_2(3)$ (see (7.2)). It is self-dual. Its weight polynomial is $W_{\mathcal{H}_8}(X, Y) = X^8 + 14X^4Y^4 + Y^8$.*

The following construction, essentially due to Alltop [1], uses knowledge of: a code, two of its subcodes and their intersection:

Theorem (10.3) (Alltop's construction XX) *Let $\mathcal{C}$ have parameters $[n, k, d]$ and $\mathcal{C}_i \subset \mathcal{C}, i = 1, 2$ with parameters $[n, k - \kappa_i, d_i]$. Let $\mathcal{C}_1 \cap \mathcal{C}_2$ have* *minimum distance* $\geq D$. *Assume there exist codes (the* **auxiliary codes***) $\mathcal{E}_i, i = 1, 2$ with parameters $[l_i, \kappa_i, \delta_i]$. Then there is a code with parameters*

$$[n + l_1 + l_2, k, min\{D, d_2 + \delta_1, d_1 + \delta_2, d + \delta_1 + \delta_2\}].$$

Proof: Apply Theorem (10.2) to the pair $\mathcal{C} \supset \mathcal{C}_1$. Consider the lengthened codes $\tilde{\mathcal{C}} \supset \tilde{\mathcal{C}_2}$. The larger of these has minimum distance $\geq min\{d_1, d+\delta_1\}$. A codeword in $\mathcal{C}_2 - (\mathcal{C}_1 \cap \mathcal{C}_2)$ obtains a nontrivial lengthening. The corresponding word in the lengthened code has then weight $\geq d_2 + \delta_1$. It follows that the minimum weight of $\tilde{\mathcal{C}_2}$ is $\geq min\{D, d_2 + \delta_1\}$. We apply (10.2) again, this time to the pair $\tilde{\mathcal{C}} \supset \tilde{\mathcal{C}_2}$, with $\mathcal{E}_2$ as auxiliary code. This yields the desired code. ∎

10.4 Lengthening and the covering radius

The following parameter is important in its own right:

Definition (10.4) (covering radius) *The* **covering radius** $\rho(\mathcal{C})$ *of the linear code $\mathcal{C}$ is the smallest number ρ such that every n-tuple in the ambient space $\mathbb{F}_q^n$ has distance $\leq \rho$ from some codeword.*

Let $\mathcal{C}$ be a code of length n and dimension k. We translate from n-space to $(n-k)$-space:

Lemma (10.3) *Let $\mathcal{C}$ be a linear q-ary code of length n and dimension k, let H be a check matrix of $\mathcal{C}$. Then $\rho(\mathcal{C})$ is the smallest number ρ such that every vector in $\mathbb{F}_q^{n-k}$ is a linear combination of ρ columns of H.*

Proof: Consider the mapping $s : x \mapsto xH^t$ from $\mathbb{F}_q^n$ to $\mathbb{F}_q^{n-k}$, which clearly is onto. Code $\mathcal{C}$ is the kernel of s. By definition the covering radius is the smallest number ρ such that for every $x \in \mathbb{F}_q^n$ we can find $y \in \mathcal{C}$ such that $d(x, y) \leq \rho$. Applying s we see that this equivalent to $s(x)$ being a linear combination of ρ columns of H.∎

What distinguishes the covering radius from the parameters considered so far is that it is not determined by internal properties of the code but by its embedding in the ambient space. Let d be the minimum distance of $\mathcal{C}$ and $\rho = \rho(\mathcal{C})$. If $\rho \geq d$, then $\mathcal{C}$ is not maximal. This means that we can pick an element $x \in \mathbb{F}_q^n - \mathcal{C}$ at distance $\geq d$ from $\mathcal{C}$. If the parameters of $\mathcal{C}$ are $[n, k, d]$, then $\mathcal{C} + \mathbb{F}_q x$ has parameters $[n, k+1, d]$. We have seen the following trivial fact:

Lemma (10.4) *Let $\mathcal{C}$ be a q-ary linear code with parameters $[n, k, d]$. If $\rho(\mathcal{C}) \geq d$, then $\mathcal{C}$ can be embedded in a code $[n, k+1, d]$.*

So when the covering radius is very large, then the code is not maximal. On the other hand, if the covering radius is small, then the code is perfect (see the Problems section). We see that the covering radius measures how tightly embedded the code is in its ambient space.

Proposition (10.2) *Let $\mathcal{C}$ be a q-ary linear code $[n, k, d]$. Then the following are equivalent:*

- $\rho(\mathcal{C}) \geq d - 1$.
- *$\mathcal{C}$ can be extended to a code $[n+1, k+1, d]$.*

Proof: Consider a check matrix H for $\mathcal{C}$. The existence of a 1-step extension is equivalent with the existence of a vector $x \in \mathbb{F}_q^{n-k}$ (to be used as $(n+1)-$st column), which is not linearly dependent of any set of $d-2$ columns of H. As x can be written in the form $x = \alpha H^t$ for some $\alpha \in \mathbb{F}_q^n$, this requirement translates into the existence of a vector $\alpha \in \mathbb{F}_q^n$, which is at distance $\geq d-1$ from $\mathcal{C}$.■

Corollary (10.2) *Let $\mathcal{C}$ be a q-ary linear code $[n, k, d]$ with covering radius $\rho < d - 1$. Then $\mathcal{C}$ can be extended to a code $[n+1, k+1, \rho+1]$. This latter code can be lengthened to yield a code $[n+d-\rho, k+1, d]$.*

Proof: The first statement follows from Proposition (10.2). The second statement is obtained by lengthening (Lemma (10.2)), using the repetition code $[d-\rho-1, 1, d-\rho-1]$ as auxiliary code and the pair $[n, k, d] \subset [n+1, k+1, \rho+1]$.■

10.5 Concatenation

Let $Q = q^h$. The method of concatenation produces a q-ary code out of a Q-ary code. In contrast to the subfield codes and trace codes considered earlier the method of concatenation relies on the choice of an additional **inner code,** which determines the step down from the large field to the small field.

Definition (10.5) (concatenation) *Let $Q = q^h$ and $\mathcal{C}$ a linear Q-ary code $[N, K, D]$ (the* **outer code***). Let further $\mathcal{E}$ be a linear q-ary code $[n, h, d]$ (the* **inner code***). Choose an $\mathbb{F}_q$-isomorphism $\alpha : \mathbb{F}_Q \longrightarrow \mathcal{E} \subseteq \mathbb{F}_q^n$. The words of the* **concatenated code** *$Concat(\mathcal{C}; \mathcal{E})$ are in bijection with the words of $\mathcal{C}$. If $(x_1, x_2, \ldots, x_n) \in \mathcal{C}$, then the corresponding word of $Concat(\mathcal{C}; \mathcal{E})$ is $(\alpha(x_1), \ldots, \alpha(x_N))$.*

Lemma (10.4) *The concatenated code $Concat(\mathcal{C}; \mathcal{E})$ of Definition (10.5) is a linear q-ary code with parameters $[nN, hk, dD]$.*

Proof: The statements concerning length and dimension are obvious. Each nonzero word of $\mathcal{C}$ has $\geq N$ nonzero entries. Each such entry yields $\geq d$ nonzero entries of the corresponding word of the concatenated code.∎

10.6 An example

We start from the hexacode $\mathcal{H}$ as constructed in Section 9.1. The geometrical description as a hyperoval in the projective plane of order four shows that $\mathcal{H}$ has precisely six one-dimensional subspaces consisting of code-words of weight six. Let $\langle v \rangle$ be one of these subspaces, which we choose as the space generated by the third line in the generator matrix of $\mathcal{H}$ from Section 9.1. We apply concatenation with the binary code $[3, 2, 2]$ as auxiliary code to the quaternary chain $\mathcal{H} \supset \langle v \rangle$. The result is a chain of binary linear codes $\mathcal{C} \supset \mathcal{D}$ with parameters $[18, 6, 8] \supset [18, 2, 12]$. Application of construction X (see Theorem (10.2)) with the all-even code $[5, 4, 2]$ as auxiliary code yields a binary code $\tilde{\mathcal{C}}$ with parameters $[23, 6, 10]$. It can be checked that there is a vector x at distance 9 from $\tilde{\mathcal{C}}$ in the ambient space $\mathbb{F}_2^{23}$. It follows that the code generated by $\tilde{\mathcal{C}}$ and x has minimum distance 9.

Theorem (10.3) *A binary code* $[24, 7, 10]$ *can be constructed as described above.*

We give a generator matrix of $[23, 7, 9]$, which has been constructed by following the steps described above. Here the concatenation $1 \mapsto 101, \omega \mapsto 011, \omega^2 \mapsto 110$ was used. The six first rows correspond to the three rows of the quaternary generator matrix from Section 9.1. The six blocks of length 3 are obtained by concatenation, the remaining five entries by construction X. The last row is the vector x at distance 9.

000	000	101	101	011	011	10001
000	000	011	011	110	110	01001
000	011	000	110	011	110	00101
000	110	000	101	110	101	00011
101	101	101	101	101	101	00000
011	011	011	011	011	011	00000
000	111	110	001	001	111	00001

1. Let $\mathcal{C}$ be a linear code $[n, k, d]_q$.
2. 1-step **shortening** produces $[n-1, k-1, d]_q$, 1-step **projection** yields $[n-1, k, d-1]_q$.
3. We speak of a 1-step **extension** if $[n+1, k+1, d]_q$ can be obtained, of a 1-step **lengthening** if $[n+1, k, d+1]_q$ can be constructed.
4. **Construction X** produces a code of the same dimension k as $\mathcal{C}$, with the minimum distance of a subcode and a larger length.
5. The **parity check** in the binary case is a special case of construction X.
6. **Construction XX** can be described as repeated application of construction X to an exact diagram.
7. The **covering radius** is another basic parameter of a code. It is related to lengthening.
8. **Concatenation** is a method to produce a code defined over a subfield of $\mathbb{F}_q$.
9. The **residual code** has parameters $[n-d, k-1, \lceil d/q \rceil]_q$. It is related to the Griesmer bound.
10. The construction of $[24, 7, 10]_2$ uses concatenation and construction X.

10.7 Problems

1. Prove that a perfect code of minimum distance $d = 2e+1$ has covering radius $\rho = e$.
2. Let $\mathcal{C}$ be a binary linear code with odd minimum distance d. Let $\mathcal{D} \subset \mathcal{C}$ be the set of code words of even weight. Prove that $\mathcal{D}$ is a subspace of codimension 1 in $\mathcal{C}$.

3. Prove that a binary linear code $[23, 7, 10]_2$ cannot exist. This shows that the code $[24, 7, 10]_2$ as constructed in Section 10.6 is best possible. Hint: Use residues (see Theorem (10.1) and Theorem (4.4)). The reasoning that led to the Hamming and Simplex codes of Chapter 7 shows that a code $[8, 5, 3]_2$ cannot exist.

4. Use concatenation with the 8-ary Denniston code of Section 9.2 as outer code and a sum-zero code as inner code to construct a binary code $[40, 9, 16]_2$.

5. Prove that a binary code $[40, 9, 17]_2$ cannot exist.
   Hint: Use residues until you are down to minimum distance 3, then use the same arguments as in the case of the Hamming codes, Chapter 7.

6. Use concatenation with the 8-ary Denniston code of Section 9.2 as outer code and our old friend the dual of the binary Hamming code as inner code. Determine the parameters $[n, k, d]_2$ of the resulting binary code and show that this code is optimal in the sense that a code $[n, k, d+1]_2$ cannot exist.

Chapter 11

The binary Golay code

In this chapter we construct the single most famous code, the binary Golay code. We start from the hexacode $\mathcal{H}$, a quaternary code. This is inspired by [18]. Along the way this will give us the opportunity to introduce the fundamental notion of a block design and to construct a particularly important block design, the large Witt design.

Definition (11.1) (binary Golay code) *Let $\mathcal{H}$ be the hexacode and $\mathbb{F}_4 = \{0, 1, \omega, \omega^2\}$. Consider $(4, 6)$-arrays $A = (A_{i,j}), i \in \mathbb{F}_4, j = 1, 2, 3, 4, 5, 6$ with binary entries $A_{i,j}$, Each such array is identified with an element of $\mathbb{F}_2^{24}$ in the natural way. We define the binary Golay code $\mathcal{G}_{24}$ as the set of those arrays, which satisfy the following two conditions:*

- *The quaternary sextuple $(\sum_{i \in F_4} A_{i,j} i), j = 1, 2, 3, 4, 5, 6$ is a codeword of $\mathcal{H}$.*
- *all the column sums of A as well as the sum of row 0 have the same parity. We shall speak of words of even and odd type in $\mathcal{G}_{24}$, respectively.*

As the conditions are linear it is clear that $\mathcal{G}_{24}$ is a binary linear code of length 24. As the most important step we determine its weight distribution. This is done by an easy counting argument. To facilitate computations we list the types of column vectors, which occur. There are 8 types according to the following dichotomies: the vector is of even or odd type, the weighted sum is either zero or nonzero in $\mathbb{F}_4$, the entry in row 0 is 0 or 1. In the following list we write these vectors as row vectors. Notation is chosen such the first entry corresponds to row 0, the second entry corresponds to the nonzero element $a \in I\!F_4$:

even/odd	weighted sum	type of vector
even	zero	$(1,1,1,1)$
even	zero	$(0,0,0,0)$
even	nonzero	$(1,1,0,0)$
even	nonzero	$(0,0,1,1)$
odd	zero	$(1,0,0,0)$
odd	zero	$(0,1,1,1)$
odd	nonzero	$(1,0,1,1)$
odd	nonzero	$(0,1,0,0)$

Let $v \in \mathcal{G}_{24}$. Denote by $\tilde{v} \in \mathcal{H}$ the quaternary vector whose entries are the weighted sums of the columns of v. Denote by α the weight of the first row of v.
If $\tilde{v} = 0$ and v is of even type, then $wt(v) = 4\alpha$. As α is even we have $wt(v) \in \{0, 8, 16, 24\}$. If $\tilde{v} = 0$ and v is of odd type, then $wt(v) = \alpha + 3(6 - \alpha) = 18 - 2\alpha$. As α is odd we have $wt(v) \in \{16, 12, 8\}$. If $wt(\tilde{v}) = 6$ and v has even type, then $wt(v) = 12$. If $wt(\tilde{v}) = 6$ and v has odd type, then $wt(v) = 3\alpha + 6 - \alpha = 6 + 2\alpha$. As α is odd we get $wt(v) \in \{8, 12, 16\}$. Remains the case that $\tilde{v}$ has weight 4. Denote by β the number of coordinates with entry 1 in the first row in the columns with weighted sum $= 0$, by γ the number of coordinates with entry 1 in the first row in the columns with weighted sum $\neq 0$. In particular $\beta + \gamma = \alpha$. If $wt(\tilde{v}) = 4$ and v has even type, then $wt(v) = 8 + 4\beta \in \{8, 12, 16\}$. In the final case when $wt(\tilde{v}) = 4$ and v has odd type we have $wt(v) = \beta + 3(2-\beta) + 3\gamma + (4-\gamma) = 10 + 2(\gamma - \beta) = 10 + 2\alpha - 4\beta$. As α is odd and $\beta \leq 2, \beta \leq \gamma$, we obtain the following weights:

$$wt(v) = \begin{cases} 12 & \text{if } \alpha = 1, \beta = 0 \\ 8 & \text{if } \alpha = 1, \beta = 1 \\ 16 & \text{if } \alpha = 3, \beta = 0 \\ 12 & \text{if } \alpha = 3, \beta = 1 \\ 8 & \text{if } \alpha = 3, \beta = 2 \\ 16 & \text{if } \alpha = 5, \beta = 1 \\ 12 & \text{if } \alpha = 5, \beta = 2 \end{cases}$$

The hard work is done. We can draw conclusions. The only weights occuring are 0,8,12,16,24. It follows from Lemma (10.2) that $\mathcal{G}_{24} \subseteq \mathcal{G}_{24}^{\perp}$. It follows

$dim(\mathcal{G}_{24}) \leq 12$. Denote by A_i the number of codewords of weight i. We have $A_0 = 1$. The all-1 codeword satisfies our conditions, hence $A_{24} = 1$. Using the description above we can determine all the A_i. In fact,

$$A_8 = \binom{6}{2} + 6 + (0+6)A_6(\mathcal{H}) + (8+2+4)A_4(\mathcal{H}) = 759,$$

$$A_{12} = \binom{6}{3} + (32 + \binom{6}{3})A_6(\mathcal{H}) + (16+4+12+4)A_4(\mathcal{H}) = 2576.$$

The presence of the all-1 vector shows that $A_{16} = A_8$. We compute $\sum_i A_i = 4096 = 2^{12}$. We have proved the following:

Theorem (11.1) *The binary Golay code $\mathcal{G}_{24}$ has parameters $[24, 12, 8]_2$. Its weight distribution is $A_0 = A_{24} = 1, A_8 = A_{16} = 759, A_{12} = 2576$. We have $\mathcal{G}_{24} = \mathcal{G}_{24}^{\perp}$.*

There is magic in the weight numbers, as we shall see. Identify the words of weight 8 with their support. This is a subset of cardinality 8 of the 24-set of coordinates. These subsets are called **octads.** We have a family of 759 octads. If any two different octads had 5 or more points in common, then the sum of the corresponding vectors would be a nonzero word of weight < 8, contradiction. It follows that the $759 \cdot \binom{8}{5}$ 5-subsets of octads are pairwise different. Here the magic happens: we have $759 \cdot \binom{8}{5} = 42,504 = \binom{24}{5}$. We conclude that each 5-subset of our 24-set of coordinates is a subset of precisely one octad. This leads us to the following important Definition:

Definition (11.2) *Let X be a finite set of cardinality v, and $\mathcal{B}$ a family of k-subsets of X. We say $(X, \mathcal{B})$ is a t-design with parameters $t - (v, k, \lambda)$ if the following holds:*
For every t-subset $T \subset X$ there are precisely λ members B of the family $\mathcal{B}$ satisfying $T \subseteq B$. The sets $B \in \mathcal{B}$ are called the **blocks** *of the design.*

We have seen examples of t-designs earlier. The projective plane $PG(2, q)$ of order q has parameters $2 - (q^2 + q + 1, q + 1, 1)$. The corresponding affine plane has parameters $2 - (q^2, q, 1)$. As every t-design is also a $(t-1)$-design the construction of t-designs becomes more demanding when t increases. We have just seen that the octads of the binary Golay code form a 5-design:

Theorem (11.2) (large Witt design) *The octads of the binary Golay code form a design with parameters* $5-(24,8,1)$. *This is known as the* **large Witt design.**

Another 5-design comes free of charge: identify the words of weights 12 in the binary Golay code with their supports. These 12-subsets are called **dodecads.** Fix a dodecad D. Let $T \subset D$ be a 5-set. We know that T is contained in a unique octad O. As $\mathcal{G}_{24}$ is self-dual, the intersection $O \cap D$ must have even cardinality. This cardinality must be 6 (if $O \subset D$, then $O+D$ would have weight 4, contradiction). We have seen the following:

Theorem (11.3) (small Witt design) *Let D be a dodecad in the binary Golay code, and $\mathcal{B}$ the family of 6-subsets $B \subset D$, which are contained in octads. Then $(D, \mathcal{B})$ is a design* $5-(12,6,1)$. *This is known as the* **small Witt design.**

11.1 Problems

1. Let $\mathcal{C} = \mathcal{RS}(4,8)$ be the self-dual 8-ary Reed-Solomon code (see Chapter 3), of parameters $[8,4,5]_8$. Apply concatenation with $[3,3,1]_2$ as inner code (see Definition (10.5)) in such a way that $\epsilon \mapsto (1,0,0), \epsilon^2 \mapsto (0,1,0), \epsilon^4 \mapsto (0,0,1)$. Show that the resulting code is a binary linear self-dual code $[24,12,8]_2$ all of whose weights are multiples of 4.
   Hint: We use the description of $\mathbb{F}_8$ from Chapter 2. Observe that $\{\epsilon, \epsilon^2, \epsilon^4\}$ is a self-dual basis.
   This construction of the binary Golay code stems from Pasquier [17].

2. Show that the Denniston arcs from Chapter 9 is a 2-design, which is embedded in the projective plane $\mathcal{P}_2(q), q = 2^a$. Determine the parameters of this design.

Chapter 12

Universal Hashing

12.1 Definitions

We mentioned in the Introduction that there is a vast range of applications of codes and orthogonal arrays, many of which are very different of the classical information-theoretic scenario. In this Chapter we study one type of applications in more detail. In Chapter 5 we used the language of arrays to introduce orthogonal arrays. We found it helpful in Section 5.2 to interpret an array in a different way: the rows are elements of a probability space, and each column is a random variable with values in the set of entries. In this chapter we will look at an array from yet another point of view: we index the columns with the elements of a set (the ground set) and interpret each row as a mapping from the ground set to the set of entries.

Definition (12.1) *An $(N; n, m)$-hash family is a multiset $\mathcal{F}$ of N functions $f : X \longrightarrow Y$, where X has cardinality n and Y has cardinality m.*

This is really nothing new. Evidently an $(N; n, m)$-hash family is formally equivalent with an (N, n)-array, with entries from a set of cardinality m.

Definition (12.2) *An $(N; n, m)$-hash family is ϵ-***universal** *for some $0 < \epsilon < 1$ if for any two distinct elements $x_1, x_2 \in X$ the number ν of functions $f \in \mathcal{F}$ such that $f(x_1) = f(x_2)$ satisfies $\frac{\nu}{N} \leq \epsilon$.*

This may be interpreted probabilistically in the following way: when different elements x_1, x_2 in the ground set are given and a mapping $f \in \mathcal{F}$ is chosen randomly according to the uniform distribution, then the probability that the function values coincide ($f(x_1) = f(x_2)$) is $\leq \epsilon$.

Definition (12.3) *An $(N; n, m)$-hash family is ϵ-***strongly universal** *for some $0 < \epsilon < 1$ if the following are satisfied:*

- *For any $x \in X$ and $y \in Y$ the number of mappings $f \in \mathcal{F}$ such that $f(x) = y$ is precisely N/m.*

- *For any two distinct elements $x_1, x_2 \in X$ and for any two (not necessarily distinct) elements $y_1, y_2 \in Y$ the number ν of functions $f \in \mathcal{F}$ such that $f(x_1) = y_1, f(x_2) = y_2$ satisfies $\frac{\nu}{N/m} \leq \epsilon$.*

The first condition says that in the corresponding array each entry occurs with the same frequency in each column. Equivalently we could say that the array is an orthogonal array of strength 1. This facilitates the probabilistic interpretation of this structure: we know that there are precisely N/m mappings f satisfying $f(x_1) = y_1$. Restrict attention to these functions only. Choose one of them at random. The second condition says that the chance of finding a functions, which satisfies $f(x_2) = y_2$, is $\leq \epsilon$. This means that ϵ is an upper bound on a conditional probability: $Pr(f(x_2) = y_2 \mid f(x_1) = y_1) \leq \epsilon$.

12.2 Authentication

Let $\mathcal{A}$ be an ϵ-strongly universal hash family. We want to show in this section how to use $\mathcal{A}$ as an authentication system. We identify $\mathcal{A}$ with the corresponding array, just as in the preceding section. Assume a source produces messages, to be sent from the originator (here called **transmitter**), to a remote **receiver**, via a public channel. In order to be able and distinguish two types of messages, those produced by the source and those (possible encrypted or otherwise changed) sent via the channel, we denote the messages produced by the source as **source states**, those sent over the channel as **messages**. We identify the source states with the columns of the array (the elements of the ground set X). Our concern is here not with secrecy (protecting the knowledge as to which source state is being transmitted) nor with error-correction (the channel may be noisy and we may wish to use an error-correcting code), but rather with **authentication**. This means that

we want to device schemes which enable the receiver to verify if the message really originated from the proper transmitter and has not been altered under way. This is where the defining properties of an ϵ-strongly universal hash family (short $\epsilon - SU$ family) will come in. We use the mappings $f \in \mathcal{A}$ as **keys**. The elements $y \in Y$ are also called **authenticators**. Assume $x \in X$ is the source state to be transmitted. Transmitter and receiver will agree, via a secure private channel, on a key $f \in \mathcal{A}$ to be used. Then the message (consisting of x, possibly in encrypted form, possibly encoded for error-correction purposes,...) will contain the **authentication tag** $f(x) = y$. Upon receiving a message which decodes to $x' \in X$, with authentication tag $y' \in Y$, the receiver will accept it as authentic if and only if $f(x') = y'$. Observe that if no problems occured, then $x' = x, y' = y = f(x)$, and the message will be accepted as authentic.
Why would we want to use an $\epsilon - SU$ hash family, and what is the meaning of ϵ? We imagine an opponent has access to the communication channel. His aim is to disturb the system, more precisely to have non-authentic messages accepted as authentic. Everything hinges upon the assumptions concerning the power of this opponent. Here we consider two types of attack:

12.2.1 The impersonation attack

It is always assumed that the opponent knows the system. His only problem is that he has no access to the secure channel, and so has no a priori information on the key used. Assume the opponent forges a message and sends it to the receiver, claiming it originated from the transmitter. He tries to maximize his chances of success. However, the first property of Definition (12.3) shows that his probability of success is precisely $1/m$, where $m = \mid Y \mid$ is the number of authenticators.

12.2.2 The substitution attack

Assume the opponent is able not only to read messages but also to alter them and send the altered version to the receiver. Knowledge of the message gives him some information on the key used. Assume he knows the source state x (we do not want to assume anything concerning secrecy). Then the secret key is one of the mappings f satisfying $f(x) = y$. How can the opponent use this knowledge to improve his chances of cheating? He wishes to have a different message, corresponding to a source state $x' \neq x$, accepted as authentic. He has to choose an authentication tag y' such that with high probability $y' = f(x')$. However, the second property of Definition (12.3) shows that his probability of success will always be $\leq \epsilon$. Observe that when $x' \neq x$ has been chosen, then the opponent's probability of success can be written as a conditional probability: $Pr(f(x') = y' \mid f(x) = y)$. Here x, x', y, y' are given ($x' \neq x$), and the probability refers to the choice of $f \in \mathcal{A}$.

12.2.3 The problem of efficiency

We have seen in the preceding subsection how an $\epsilon - SU$ hash family can be interpreted as an authentication system (often called **authentication code**) with the property that an opponent's probability of success with a substitution attack in the sense of Subsection 12.2.2 is provably limited by ϵ. It should be observed that for each transmission of a source state a new key has to be generated at random and this information transmitted to the receiver beforehand, via a secure channel. In order to get a system, which can be used in practice, we have to limit the amount of time and space required by authentication. If for example there are $N = 2^{100}$ keys and $m = 2^{40}$, then the authenticator tag will have 40 bits and the information on the key, to be sent via the secure channel, will be a bit string of length 100. It would not make much sense to use such a system when the number of messages is $n = \mid X \mid = 2^{100}$. We cannot afford to spend half of the transmission time for authentication purposes. This leads to the problem of efficiency. The parameters n (the number of source states) and ϵ (the level of security) can be thought of as given. We want to construct such authentication systems, equivalently $\epsilon - SU$ hash families, such that the number N of keys and the number m of authenticators are as small as possible. The most reasonable measure seems to be $log_2(N) + log_2(m) = log_2(Nm)$. This is the parameter we wish to minimize, where n and ϵ are given.

12.3 Basic theory

In the preceding section we gave an interpretation of $\epsilon - SU$ hash families as authentication codes. We will come back to more applications later. The ϵ-**universal** hash families (short $\epsilon - U$ hash families) have their own appli-

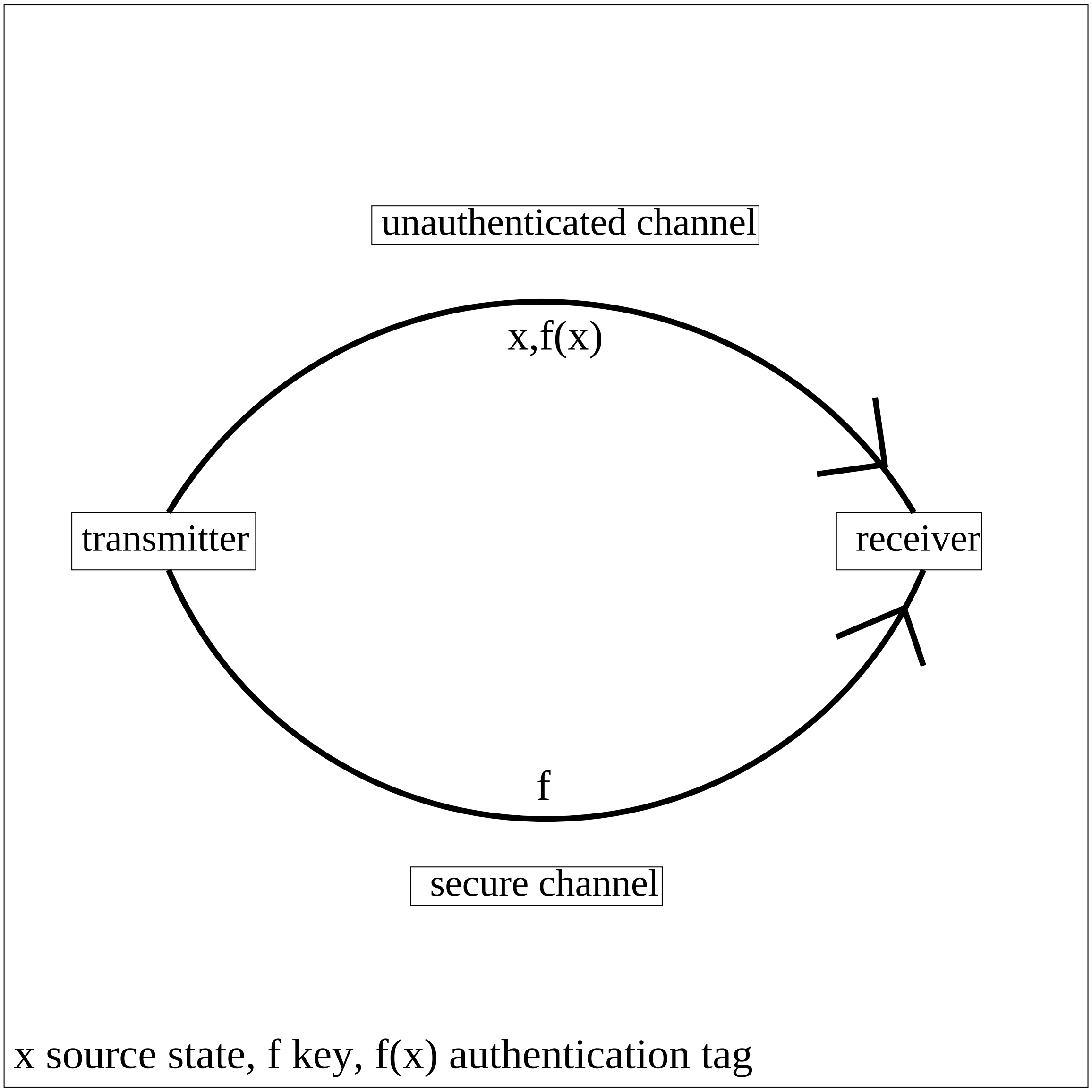

Figure 12.1: Authentication with ϵ-SU

cations. In this section we will interpret these structures from the point of view of coding theory and discuss methods of construction. It will turn out that $\epsilon - U$ hash families are codes in disguise. Moreover they are important ingredients in the construction of $\epsilon - SU$ hash families, and these latter are generalizations of orthogonal arrays of strength 2.
Let $\mathcal{A}$ be an $\epsilon - U$ hash family, written as array. Consider the columns as words of an m-ary code, of length N. Let x, x' be two different such codewords (=columns=elements of the ground set). Then the defining property of Definition (12.2) can be interpreted in terms of the Hamming distance $d(x, x')$. It says $\frac{d(x,x')}{N} \geq 1 - \epsilon$. If d is the minimum distance of our code, then the defining property of an $\epsilon - U$ hash family can equivalently be expressed as $\frac{d}{N} \geq 1 - \epsilon$, where N is the length of the code. In coding theory the quantity d/N is known as the **relative minimum distance** of a code. We have seen the following:

Theorem (12.1) *Let $\mathcal{A}$ be an $(N; n, m)$-hash family as in Definition (12.1). Then the following are equivalent:*

- *$\mathcal{A}$ is an $\epsilon - U$ hash family.*
- *The columns of $\mathcal{A}$ form the words of an m-ary code of length N with relative minimum distance $\frac{d}{N} \geq 1 - \epsilon$.*

We see that $\epsilon - U$ hash families are just codes in disguise. Any application of an $\epsilon - U$ hash family is an application of codes. We turn to $\epsilon - SU$ hash families. As the defining property is in terms of pairs of columns of the corresponding array it is natural to compare with orthogonal arrays of strength 2.

Lemma (12.1) *Let $\mathcal{A}$ be an $(N; n, m)$-hash family, which is $\epsilon - SU$. Then $\epsilon \geq 1/m$. Equality holds if and only if $\mathcal{A}$ is an orthogonal array of strength 2.*

Proof: Let x, y, x' be given as before. As y' varies over the m-set Y we see that the maximum of the conditional probabilities $Pr(f(x') = y' \mid f(x) = y)$ must be $\geq 1/m$. It follows that $\epsilon \geq 1/m$. In the case of equality it must be the case that for all x, y, x', y' as before $(x, x' \in X, x' \neq x; y, y' \in Y)$ the number of mappings f satisfying $f(x) = y, f(x') = y'$ must be a constant, say

λ. We compare with the definition of an orthogonal array (Definition (5.5)) and see that we have an orthogonal array of strength 2. ∎

Lemma (12.2) *Let $\mathcal{A}$ be an $(N; n, m)$-hash family, which is $\epsilon - SU$. Then $\mathcal{A}$ is $\epsilon - U$.*

The proof of this elementary fact is left to the reader (see the Problems section). It follows that $\epsilon - SU$ families satisfy stronger conditions than $\epsilon - U$ families. Lemma (12.1) shows that we may use orthogonal arrays of strength 2 as authentication schemes. However, the problem of efficiency arises. The Bose-Bush bound (see the Problems section of Chapter 5) shows that the number N of rows (keys) certainly exceeds the number n of columns (source states). This shows that OA of strength 2 are extremely inefficient when used as authentication schemes. Next we describe the hitherto most efficient construction technique for $\epsilon - SU$ hash families:

Theorem (12.2) (Stinson composition) *Let $\mathcal{A}_1$ be an $(N_1; n, n')$-hash family, which is $\epsilon_1 - U$ and $\mathcal{A}_2$ be an $(N_2; n', m)$-hash family, which is $\epsilon_2 - U$. The* **composition** *$\mathcal{A} = \mathcal{A}_2 \circ \mathcal{A}_1$ is defined as the multiset of compositions of mappings $f = f_2 \circ f_1$, where $f_i \in \mathcal{A}_i$. Then $\mathcal{A}$ is an $(N_1 N_2; n, m)$-hash family, which is $\epsilon - SU$, where $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_1 \epsilon_2$.*

Proof: As $\mathcal{A}_i$ is a multiset of N_i mappings $(i = 1, 2)$ it is clear that the multiset $\mathcal{A}$ of all compositions consists of $N_1 N_2$ mappings. We have that $\mathcal{A}$ is an $(N_1 N_2; n, m)$-hash family. The first condition of Definition (12.3) is satisfied. In order to prove the main condition we use the language of probabilities. So let x_1, x_2 in the ground set $(x_1 \neq x_2)$ and z_1, z_2 in the m-set be given. Write $f \in \mathcal{A}$ as $f = f_2 \circ f_1$, let $y_1 = f_1(x_1), y_2 = f_1(x_2)$ and $P = Prob(f(x_1) = z_1 \mid f(x_2) = z_2)$. We want to show that $P \leq \epsilon_1 + \epsilon_2 - \epsilon_1 \epsilon_2$. Consider at first the case $z_1 \neq z_2$. We have $P \leq Prob(y_1 \neq y_2)\epsilon_2 < \epsilon_2$. The case $z_1 = z_2$ is harder. We have

$$P \leq Prob(y_1 \neq y_2)\epsilon_2 + Prob(y_1 = y_2).$$

Let $\alpha = Prob(y_1 = y_2)$. We have $P \leq \alpha + (1 - \alpha)\epsilon_2$, which we denote as $h(\alpha)$. As $h'(\alpha) = 1 - \epsilon_2 \geq 0$ and $\alpha \leq \epsilon_1$ by the defining property of $\mathcal{A}_1$ we get $P \leq h(\epsilon_1)$, which is our claim.■

In the next section we will show that this construction technique can be used to obtain rather efficient authentication codes. It will turn out that by increasing the security parameter ϵ only slightly above its theoretical minimum $\epsilon - SU$ hash families can be obtained with a dramatically smaller number of mappings (keys). We address this effect as the Wegman&Carter-effect as this effect has first been noticed in their joint papers [11, 26].

12.4 Construction of strongly universal hash families

Theorem (12.3) *Let q be a prime-power, let k, α, β be natural numbers, $\alpha \geq \beta$. Then there is an $(q^{\alpha+\beta}; q^{k\alpha}, q^{\beta})$ hash family, which is $\frac{k}{q^{\beta}} - SU$.*

Proof: Fix a surjective $\mathbb{F}_q-$linear mapping

$$\phi : \mathbb{F}_{q^{\alpha}} \longrightarrow \mathbb{F}_{q^{\beta}}.$$

We form an array $\mathcal{A}$ with $q^{k\alpha}$ columns, $q^{\alpha+\beta}$ rows, and with entries in $\mathbb{F}_{q^{\beta}}$: The columns are indexed by the polynomials $p(X)$ with coefficients in $\mathbb{F}_{q^{\alpha}}$, of degree $\leq k$ and with constant term =0. The rows are indexed by the pairs (u, v), where $u \in \mathbb{F}_{q^{\alpha}}, v \in \mathbb{F}_{q^{\beta}}$. The entry in row (u, v) and column $p(X)$ is

$$\phi(p(u)) + v.$$

It is clear that each column of $\mathcal{A}$ contains each element of $\mathbb{F}_{q^{\beta}}$ with frequency q^{α}. Fix two different columns $p_1(X), p_2(X)$ and two not necessarily different elements $y_1, y_2 \in \mathbb{F}_{q^{\beta}}$. We have to count the rows (u, v), which satisfy the following:

$$\phi(p_1(u)) + v = y_1 \tag{12.1}$$
$$\phi(p_2(u)) + v = y_2. \tag{12.2}$$

We subtract and use the linearity of ϕ. This leads to the equivalent equation

$$\phi((p_1 - p_2)(u)) = y_1 - y_2.$$

Each solution u of this equation gives us a uniquely determined row (u, v). Put $M = \phi^{-1}(\alpha_1 - \alpha_2)$. We have $|M| = q^{\alpha-\beta}$. Observe that $p_1(X) - p_2(X)$ is a nonconstant polynomial of degree $\leq k$. It follows that the number of solutions is $\leq k \cdot q^{n-m}$. ∎

Consider the special case $k = 1$ in Theorem (12.3). We have $\epsilon = q^{-\beta} = 1/m$. It was observed in Lemma (12.1) that this characterizes OA of strength 2.

Corollary (12.1) *The special case $k = 1$ of Theorem (12.3) yields orthogonal arrays with parameters $OA_{q^{\alpha-\beta}}(2, q^\alpha, q^\beta)$.*

Let us apply Stinson composition Theorem (12.2) with the OA of Corollary (12.1) in the role of $\mathcal{A}_2$. We know that $\mathcal{A}_1$, an $\epsilon_1 - U$ hash family, is equivalent with an error-correcting code. Let us use Reed-Solomon codes.

Lemma (12.3) *The Reed-Solomon code $\mathcal{RS}(k, q)$ as in Definition (3.1) yields an $(q; q^k, q)$-hash family, which is $\frac{k-1}{q}$-universal.*

Proof: Comparing with Theorem (12.1) we see that the parameters N, n, m are as given. As the Reed-Solomon code has minimum distance $d = q-(k-1)$ it follows that the $\epsilon = 1 - \frac{d}{q} = \frac{k-1}{q}$. ∎

Stinson composition, using Reed-Solomon codes and the OA from Corollary (12.1) (which are themselves very closely related to Reed-Solomon codes) as ingredients, yields an interesting family of strongly universal hash families:

Theorem (12.4) *Let q be a prime-power and α, β natural numbers such that $\alpha \geq \beta$. Then there is an $(q^{2\alpha+\beta}; q^{\alpha q^{\alpha-\beta}}, q^\beta)$-hash family, which is $2q^{-\beta} - SU$.*

Proof: $\mathcal{RS}(q^{\alpha-\beta}, q^\alpha)$ yields a $(q^\alpha; q^{\alpha q^{\alpha-\beta}}, q^\alpha)$-hash family, which is $\epsilon_1 - U$. Here $\epsilon_1 = (q^{\alpha-\beta} - 1/q^\alpha) < q^{-\beta}$. An OA from Corollary (12.1) yields a $(q^{\alpha+\beta}; q^\alpha, q^\beta$-hash family, which is $q^{-\beta} - U$. The composition has $q^{2\alpha+\beta}$ members. Its ϵ-value as a strongly universal family is $\epsilon \leq \epsilon_1 + q^{-\beta} < 2q^{-\beta}$. ∎

Here is an example to illustrate:

Example (12.1) *Choose* $q = 2, \beta = 20, \alpha = 32$. *Theorem (12.4) yields a* $(2^{84}; 2^{2^{17}}, 2^{20})$*-hash family, which is* $2^{-19} - SU$. *In the language of authentication (Section 12.2) we see that we can authenticate a* 2^{17}*-bit source with* 20 *authentiicator bits and 84 key bits such that the deception probability is limited to* 2^{-19}. *This illustrates how dramatic the Wegman&Carter-effect is. Had we insisted on a deception probability of* 2^{-20}, *while still using only 20 authenticator bits, then we would have been forced to use an* OA *of strength 2 by Lemma (12.1), leading to more than* 2^{17} *key bits because of the Bose-Bush bound, as mentioned earlier.*

12.5 An elementary application: Associative Memories

In all the applications the domain X is chosen to be considerably larger than the range Y. In computer science functions $f : X \longrightarrow Y$ are often called **hash functions** if the intention is to partition (to hash) X into small pieces (those with the same value under f). Mathematically one could speak of partition functions. Suitably chosen hash functions may be used in search algorithms in the following way: whenever some $x \in X$ is stored, then we also store the value $f(x)$ (the hash value). If one wants to check if some $x \in X$ has been stored already, then one computes the hash value at first and searches only through the shorter list of those elements of X with this hash value. This is a very innocent application. For example, imagine a compiler, which in a table of symbols lists all identifiers that have been used in a certain program. The number of all possible identifiers is very large, about $1.62 \cdot 10^9$ in $FORTRAN$. One can use an $\epsilon - SU$ class $\mathcal{A}$ with the set of possible identifiers as domain. Choose some $f \in \mathcal{A}$ at random and use it as hash function. Assume x is an identifier. We want to check if it is new. Instead of searching through our huge set compute $f(x)$ at first and search through the list of all x' satisfying $f(x') = f(x)$. Although only one hash function f is really used this may be seen as an application of universal hashing as the properties of the hash family determine the expected behaviour of the algorithm.

12.6 Multiple authentication with counters

A drawback of our method as given in Section 12.2 is that each key may be used only once. Suppose we want to economize on the number of key bits by using each key a certain number of times for the authentication of several messages. It would certainly be possible to use OA of higher strength. However, the problem of efficiency (see Subsection 12.2.3) imposes severe restrictions: the Bose-Bush bound shows that the number of keys would be much too big. Here is a method to get around this problem:
Let $\mathcal{A}$ be an $(N; n, m)-$hash family, which is $\epsilon - SU$. Impose the structure of a group on the set of entries: $Y = \mathbb{Z}/m\mathbb{Z}$. We wish to be able and authenticate up to w source states with the same key. The keys are tuples $(f, y_1, \ldots, y_{w-1})$, where f is chosen randomly from $\mathcal{A}$ and the y_i are chosen independently and uniformly at random from Y. Let $(x_1, x_2, \ldots, x_w)$ be a tuple of source states ($x_i \in X$, pairwise different), to be authenticated with the same key. The messages have the form (i, x_i). Here i is the **counter**. The authentication tag is then $f(x_1)$ if $i = 1$; it is $f(x_i) + y_{i-1}$ if $i > 1$. Assume the opponent observes up to w messages, which have been authenticated with the same key. He uses this information to alter the last of these messages in such a way that the source state is replaced by some $x \notin \{x_1, x_2, \ldots, x_w\}$, his aim being to have x accepted as authentic. It is an easy exercise to check that his probability of success is bounded by ϵ.

12.7 Problems

1. Prove Lemma (12.2).

Chapter 13

The Gilbert-Varshamov bound

Our first results will follow from simple counting arguments. Recall the volume $V(i,n) = V_q(i,n) = \sum_{j=0}^{i} \binom{n}{j}(q-1)^j$ of the ball of radius i as introduced in Definition (4.1). Let $\mathcal{C}$ be a q-ary code with parameters $[n, k-1, d]$. As $\mathcal{C}$ has q^{k-1} elements it follows that if $q^{k-1}V(d-1,n) < q^n$, then there is a vector $v \in \mathbb{F}_q^n$, which has distance $\geq d$ from every code-word $\in \mathcal{C}$. This leads to the Gilbert-Varshamov bound:

Theorem (13.1) (Gilbert-Varshamov bound) *If $V(d-1,n) < q^{n-k+1}$, then a q-ary linear code with parameters $[n,k,d]_q$ exists.*

As usual it turns out to be profitable to start from the dual setting:

Theorem (13.2) *If $V(d-2,n-1) < q^{n-k}$, then a q-ary linear code with parameters $[n,k,d]_q$ exists. Moreover every code $[n-1,k-1,d]$ can be embedded in a code $[n,k,d]$.*

Proof: Let $\mathcal{A}$ be a linear subspace of dimension $n-k$ of $\mathbb{F}_q^{n-1}$, which is an orthogonal array of strength t, and let A be a generator matrix of $\mathcal{A}$. We wish to add an additional column to A such that the resulting subspace of $\mathbb{F}_q^n$ still is an orthogonal array of strength t. The columns which do not do the job are precisely those vectors in $\mathbb{F}_q^{n-k}$, which can be written as linear combinations of at most $t-1$ columns of A. The number of such linear combinations is at most $\sum_{i=0}^{t-1}\binom{n-1}{i}(q-1)^i$. This number happens to equal $V(t-1,n-1)$. Thus, if $V(t-1,n-1) < q^{n-k}$, then our orthogonal array possesses a 1-step lengthening. Considering duals we see that we have proved the following: if there is a code $\mathcal{C}$ with parameters $[n-1, k-1, d]$ and if $V(d-2, n-1) < q^{n-k}$, then $\mathcal{C}$ can be extended to a code $[n, k, d]$. Just as in the case of the Gilbert-Varshamov bound it is easy to see by induction that the condition of the existence of an $[n-1, k-1, d]$ is not needed.$\blacksquare$

Lemma (13.1) *Theorem (13.2) is always better than the original Gilbert-Varshamov bound Theorem (13.1).*

Proof: Fix parameters $[n, k, d]$. We ask under which conditions the two Theorems in question guarantee the existence of a q-ary code with these parameters. This makes it clear that our claim is equivalent with the inequality $qV(d-2, n-1) < V(d-1, n)$. It is easy to see, that this equality holds. In fact, consider the $V(d-2, n-1)$ vectors of length $n-1$ and weight $\leq d-2$. Adding a coordinate and extending each of these vectors in all q possible ways yields $qV(d-2, n-1)$ different (but obviously not all) vectors of length n and weight $\leq d-1$. $\blacksquare$

Using both Theorem (13.2) and Theorem (10.2) yields another code construction:

Theorem (13.3) *Assume $V(d-2, n-1) < q^{n-k}$. If there exist codes $[n-i, k-i, d+\delta]$ and $[e, i, \delta]$, then a code $[n+e, k, d+\delta]$ can be constructed.*

Proof: The assumptions of our Theorem show that the code $[n-i, k-i, d+\delta]$ can be embedded in a code $[n, k, d]$. Application of Theorem (10.2) to this pair of codes leads to the conclusion of the Theorem.$\blacksquare$

Remarks

The foundations of what are now called information theory and coding theory were laid in an extremely influential paper of Shannon's [21], in 1948. A basic theorem proved by Shannon may best be described as follows: if we are willing to use very long codes then very good codes always exist.
An important asymptotical question is the following: let $R = k/n$ be the **information rate** and $\delta = d/n$ the **relative minimum distance** of linear

codes $[n, k, d]_q$. For which pairs (R, δ) can we find infinite families of codes $[n_i, k_i, d_i]_q$ such that $lim_{i \mapsto \infty} k_i/n_i \geq R$ and $lim_{i \mapsto \infty} d_i/n_i \geq \delta$? We will come back to this question in Chapter 15.

Chapter 14

The basics of information theory

14.1 Shannon entropy: definition and meaning

Definition (14.1) (entropy) *Let $p_i, i = 1, \dots, n$ be a probability distribution. The* **Shannon entropy** *H is defined by*

$$H(p_1, p_2, \dots, p_n) = -\sum_{i+1}^{n} p_i \cdot log(p_i).$$

Here as always in information theory the logarithm is taken base 2. We put $0 \cdot log(0) = 0$. Imagine a source, which at every unit of time emits one of n symbols, called letters, according to the given probability distribution. That is, the probability for letter i to be chosen is p_i. We claim that the entropy function is a measure for the average amount of information per letter generated by this source. This will be made explicit and proved in the remainder of this section.

Theorem (14.1) *Let $H_n : \mathbb{R}_+^n \longrightarrow \mathbb{R}_+ (n = 2, 3, \dots)$ be a family of functions satisfying the following properties:*

- *Each H_n is continuous.*
- *$H_2(1/2, 1/2) = 1$ (one bit).*

- *There is a continuous monotonely increasing function $A(x) : I\!R_+ \longrightarrow I\!R_+$ such that $A(n) = H_n(1/n, \ldots, 1/n)$.*

- $$H_n(p_1, p_2, \ldots, p_n) = H_2(p_1, 1-p_1) + (1-p_1) \cdot H_{n-1}(\frac{p_2}{1-p_1}, \ldots, \frac{p_n}{1-p_1}).$$

  Here the numbers p_i describe a probability distribution.

Then $H_n(p_1, \ldots, p_n)$ is the entropy function.

The whole idea behing the definition of entropy is in the last property. It says that the average amount of information contained in the choice between certain probabilities according to a given distribution is not changed when the information is revealed in various steps. We shall use this property and obvious consequences obtained by repeated use of it:
The last rule implies $A(s^m) = A(s^{m-1}) + A(s)$ for naturals s, m. By induction $A(s^m) = mA(s)$. Let naturals s, t be given. For every natural number n choose m such that $s^m \leq t^n < s^{m+1}$. Because of monotonicity and the above rule we have $mA(s) \leq nA(t) < (m+1)A(s)$, hence $m/n \leq A(t)/A(s) < (m+1)/n$. These relations also hold for the $log-$function instead of $A(x)$. By letting $n \longrightarrow \infty$ we get $A(t)/A(s) = log(t)/log(s)$. Put $s = 2$. This shows $A(t) = log(t)$.
Consider rational probabilities $p_i = m_i/m, i = 1, 2, \ldots, n$. Using an obvious consequence of the central property we get $A(m) = H_n(p_1, p_2, \ldots, p_n) + \sum_i p_i \cdot A(m_i)$. Using the fact that $A(x) = log(x)$ and solving for H_n we get the desired expression, in the case of rational probabilities. Finally use continuity.■

14.2 Basic properties of Shannon entropy

A basic tool in proving the basic properties of Shannon's entropy function is Jensen's inequality.

Definition (14.2) *Let $f(x)$ be a real function, defined on an open interval I. f is* **concave** *if for every $x_0 \in I$ there is a line L through $(x_0, f(x_0))$, which is never under the graph of $f(x)$.*

Let $g(x) = f(x_0) + \lambda(x - x_0)$ be the equation of this line. By definition of concavity $f(x) \leq f(x_0) + \lambda(x - x_0)$ for all $x \in I$. Let $x_1 \leq x_2 \ldots \leq x_n$ be points in I and $p_1, \ldots, p_n$ be probabilities of a probability space. Using the inequalities above yields

$$\sum_i p_i f(x_i) \leq f(x_0) + \lambda(\sum_i p_i x_i - x_0).$$

Choose in particular $x_0 = \sum_i p_i x_i$, observe that $x_0 \in I$.

Theorem (14.2) (Jensen's Inequality) *Let $f(x)$ be concave on the open real interval I, let $x_1 \leq x_2 \ldots \leq x_n$ be points in I and $p_1, \ldots, p_n$ be probabilities of a probability space. Then*

$$\sum_i p_i f(x_i) \leq f(\sum_i p_i x_i).$$

Equality holds if and only if $f(x)$ is a linear function (its graph is a straight line) on the interval $[x_1, x_n]$.

The proof for the case of equality is left as an exercise (see the Problems section).
Clearly the *log*-function is concave on the interval $(0, \infty)$. Choose positive numbers x_i and a probability distribution p_i. An application of Jensen's inequality yields the famous inequality between the arithmetic and the geometric mean:

Theorem (14.3) (arithmetic and geometric mean) *Let $x_i > 0, p_i \geq 0, i = 1, 2, \ldots, n$, where $\sum_i p_i = 1$. Then the following holds:*

$$\sum_{i=1}^{n} p_i \cdot x_i \geq \prod_{i=1}^{n} x_i^{p_i}.$$

Equality holds if and only if all the x_i with $p_i \neq 0$ are equal.

Another application of Jensen's inequality to the *log*-function will be used to establish a key property of the entropy:

Proposition (14.1) *Let p_i and q_i be positive numbers $(i = 1, 2, \dots, n)$ such that $\sum_i p_i = 1$. Put $\sum_i q_i = \alpha$. Then the following holds:*

$$\sum_i p_i \cdot log(1/p_i) \leq \sum_i p_i \cdot log(1/q_i) + log(\alpha).$$

Equality holds if and only if $q_i = \alpha p_i$ for all i.

Proof: Apply Jensen's inequality to $x_i = q_i/p_i$. Observe $\sum_i p_i x_i = \sum_i q_i = \alpha$.■

In case $\alpha = 1$, when the q_i form another probability distribution, this yields a first property of the entropy function: consider the expression $\sum_i p_i \cdot log(1/q_i)$, where the p_i form a fixed probability distribution (q_i) varies through the probability distributions. The Proposition says that this expression is minimized when $q_i = p_i$, hence by the entropy.

14.2.1 Joint entropy and conditional entropy

Every probability space has its entropy. As every **random variable** (mapping defined on a probability space) induces a probability distribution, we can also speak of the entropy of a random variable. Let X and Y be random variables with probability distributions $p_i = Pr(X = i)$ and $q_j = Pr(Y = j)$, respectively. Consider further the joint distribution of X and Y, given by the probabilities $p(i, j) = Pr(X = i \text{ and } Y = j)$. Observe that the joint distribution carries more information than the distributions of X and Y taken together. In fact, we may arrange the probabilities $p(i, j)$ in a matrix with rows indexed by i and columns indexed by j. The row sums are the p_i, the column sums are q_j. The other way around it is clear that the matrix is far from being determined by its row- and column sums. Recall the important notion of statistical independence: two random variables X and Y as above are **(statistically) independent** if $p(i, j) = p_i \cdot q_j$ always holds. The meaning behind this notion is that in the case of independent random variables information about the value of one of them gives no information whatsoever on the value of the other variable. We can compare the entropies $H(X)$ and $H(Y)$ to the joint entropy $H(X, Y)$. In fact, there are more entropies around. Consider the **conditional probability**

$$p(i|j) = p(i, j)/q_j.$$

This is the probability that $X = i$ if it is already known that $Y = j$. Observe that X and Y are independent if and only if $p(i|j) = p_i$ always holds. For every fixed value j of Y the $p(i|j)$ form a probability distribution. Denote the corresponding entropy by $H(X|Y = j)$. It is natural to define a conditional entropy $H(X|Y)$ as the weighted sum of these expressions, the weights being the probabilities q_j.

Definition (14.3) (conditional Shannon entropy) *The conditional entropy $H(X|Y)$ is defined by*

$$H(X|Y) = \sum_j q_j \cdot H(X|Y = j) = -\sum_{i,j} p(i,j) \cdot log(p(i|j)).$$

Here i, j run through the values of X and Y, respectively.

Observe that the conditional entropy is a new notion. It is not the entropy of a probability space. There is no probability space $X|Y$.

14.2.2 The main theorem

The main theorem shows that our notions cooperate in exactly the way our interpretation suggests:

Theorem (14.4) *Let X and Y be random variables defined on finite probability spaces. Then the following relations hold for the entropies:*

- $H(X, Y) \leq H(X) + H(Y)$, *with equality if and only if X and Y are independent.*
- $H(X, Y) = H(X) + H(Y|X)$.
- $H(X) \geq H(X|Y)$, *with equality if and only if X and Y are independent.*

Proof: Most interesting is the proof of the first statement. If we write out the definitions we see that we have to prove the following statement: $\sum_{i,j} p(i,j) \cdot log(p(i,j)) \geq \sum_{i,j} p(i,j) \cdot log(p_i q_j)$. We can apply Proposition 1 in case $\alpha = 1$. It also follows from this Proposition when equality occurs.
The second statement is a trivial formal manipulation, the third statement follows from the others: $H(X) + H(Y) \geq H(Y, X) = H(Y) + H(X|Y)$.■

Recall that we interpret $H(X)$ as a measure for the average amount of information per letter of a source operating with the corresponding probabilities. The first property says that the information contained in X and Y taken together is at most the sum of the individual informations. The second statement says that the joint information given by X and Y together can be written as the sum of the information given by X and the information given by Y when X is known. The third property says that knowledge on Y cannot increase our uncertainty about X. Exactly then will this knowledge not help when X and Y are independent.
Finally, we come to a natural notion, which is not to be found in Shannon's classical papers, mutual information:

Definition (14.4) (mutual information) $I(X, Y) = H(X) - H(X|Y)$.

Observe that this notion is symmetric in X and $Y : I(X, Y) = H(X, Y) - H(Y|X) - H(X|Y) = I(Y, X)$. Does this notion agree with our interpretation? $H(X)$ is the uncertainty about X, and $H(X|Y)$ is the remaining uncertainty about X once Y is known. We should therefore interpret the difference as the information Y has disclosed about X. For future use we record the concrete formula for the mutual information:

Lemma (14.1) $I(X, Y) = \sum_{i,j} p(i, j) log(\frac{p(i|j)}{p_i})$.

14.3 Another interpretation of entropy: the uniform model

The **law of large numbers** provides another interpretation of the entropy function. Consider a probability space with probabilities $p_1, p_2, \ldots, p_n$. As before consider this space as a source of information, where at every point of time one of n symbols is generated according to the probability distribution. Consider signals of length N generated by this source. Let h_i be the frequency of letter $i \in \{1, 2, \ldots, n\}$. The **law of large numbers** asserts the following: For every pair $\epsilon, \delta > 0$ there is a $N_0 = N_0(\epsilon, \delta)$ such that for every $N \geq N_0$ the set of signals of length N can be partitioned in two parts:

- a set of total probability $< \epsilon$, and

- a set all of whose elements satisfy

$$|h_i/N - p_i| < \delta.$$

This means that if we pick a long signal at random, then we are almost sure that the relative frequencies of the letters are close to the probabilities. If S is such a signal of length N, then its probability is $p = Pr(S) = \prod_i p_i^{h_i}$. It follows $log(p) = \sum_i h_i \cdot log(p_i)$, hence $log(p)/N = \sum_i (h_i/N) \cdot log(p_i)$. If S is one of the probable signals, then this expression is close to $\sum_i p_i \cdot log(p_i) = H(p_1, \dots, p_n)$. This is another interpretation of the entropy function:
If a source generates letters from a set of n letters according to a fixed probability distribution p_i, then we are almost sure that a signal of large length picked at random has probability p such that $\frac{log(1/p)}{N}$ is close to $H(p_1, \dots, p_n)$. For considerations where only signals of large lengths are considered and events of small probability can be disregarded it is therefore possible to describe the output of the source as follows:
there are $2^{H \cdot N}$ signals of length N, and each such signal occurs with the same probability $2^{-H \cdot N}$. We call this the **uniform model.**
We can use this to exhibit a close relationship between the binomial coefficients $\binom{n}{m}$ and the binary entropy function $h(p) = H(p, 1-p)$, as follows: let $\sigma = (\sigma_1, \sigma_2, \sigma_k)$, where each σ_i is chosen from the $\binom{n}{m}$ binary sequences with m entries 1, according to the uniform distribution (each such sequence with the same probability $1/\binom{n}{m}$). We can see σ as a binary sequence of length nk. Compare with the binary sequences of length nk, where in each coordinate 1 is chosen with probability $p = n/m$. Fix p, let $m, k \longrightarrow \infty$. By the uniform model the sets of typical sequences of these two families agree. The probability of a typical signal is $1/\binom{n}{m}^k$, and it is asymptotically $1/2^{nkh(p)}$. It follows that asymptotically $\binom{n}{m} \sim 2^{nh(m/n)}$.

14.4 Problems

1. Prove the statement concerning the case of equality in Jensen's inequality (Theorem (14.2)).

Chapter 15

Asymptotic bounds

Let $\mathcal{C}$ be a code $[n,k,d]_q$. Call $\delta = d/n$ the **relative minimum distance** and $R = k/n$ the **information rate.** In the case of not necessarily linear codes the information rate is defined more generally as $R = log_q(\mid \mathcal{C} \mid)/n$. Let us call a pair (δ, R) **asymptotically reachable** by linear q-ary codes if there is an infinite family of such codes with length $n_i \longrightarrow \infty$, relative minimum distances δ_i and information rates R_i such that $lim_i \delta_i \geq \delta$ and $lim_i R_i \geq R$. The bounds of Chapter 4 yield upper bounds on asymptotically reachable pairs (δ, R). The Singleton bound shows that $\delta + R \leq 1$. The Plotkin bound yields a stronger result: $R = 0$ if $\delta \geq (q-1)/q$ and $R \leq 1 - q\delta/(q-1)$ if $\delta \leq (q-1)n/q$. In other words, asymptotically reachable points (δ, R) are under the line joining the points $(0,1)$ and $(\frac{q-1}{q}, 0)$. Consider the sphere packing bound Theorem (4.2):$\sum_{i=0}^{e} \binom{n}{i}(q-1)^i \leq q^{n-k}$. Take base 2 logarithms and divide by n. The right hand side is then $1 - R$. Take the limit, where $n \longrightarrow \infty, d/n \longrightarrow \delta$. The dominating term in the sum on the left is $i = e$. As $lim(log_q(e)/n) = 0$ we can replace the left side by $\binom{n}{e}(q-1)^e$. We have seen in Section 14.3 that $\binom{n}{e} \sim 2^{nh(e/n)}$. Here $h(p) = H(p, 1-p)$ (see Chapter 14) and we observe that $lim(h(e/n)) = h(\delta/2)$. We obtain the **asymptotic sphere packing bound**

$$R \leq 1 - \{\frac{\delta}{2} log_q(q-1) + log_q(2)h(\delta/2)\}.$$

Along the same lines the Gilbert-Varshamov bound Theorem (13.1) yields a lower bound. Observe that the term $i = d-1$ will be asymptotically dominating in the sum on the left side provided $\delta \leq (q-1)/q$. We obtain the **asymptotic Gilbert-Varshamov bound:**

If $\delta \leq (q-1)/q$ and $R \leq 1-h(\delta)-\delta log_q(q-1)$, then (δ, R) is asymptotically reachable.

Chapter 16

Codes from field extensions

16.1 Introduction and prerequisites

The construction of many good linear codes involves the following step: one constructs a code $\mathcal{C}$, which is linear over an extension field $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$. A linear q-ary code is then derived either as a the trace code or as the subfield code of $\mathcal{C}$. The facts collected in this section are not hard to understand or to prove. Nevertheless they are extremely important and should be studied with care.

Definition (16.1) *Let $F = \mathbb{F}_{q^r}|\mathbb{F}_q$ and let $\mathcal{C}$ be an F-linear code. Denote the trace by $tr = tr : F \longrightarrow \mathbb{F}_q$. The* **trace code** *$tr(\mathcal{C})$ is defined as the set of all $(tr(x_1), tr(x_2), \ldots, tr(x_n))$ where $(x_1, x_2, \ldots, x_n) \in \mathcal{C}$. The* **subfield code** *$\mathcal{C}_{F_q}$ consists of the codewords of $\mathcal{C}$ all of whose entries happen to be in $\mathbb{F}_q$, formally $\mathcal{C}_{F_q} = \mathcal{C} \cap \mathbb{F}_q^r$.*

It is clear that $tr(\mathcal{C})$ and $\mathcal{C}_{F_q}$ are linear q-ary codes. The relationship between them is clarified by the following Theorem, which is attributed to Delsarte:

Theorem (16.1) *In the situation of the preceding Definition the following holds:*

$$tr(\mathcal{C})^{\perp} = (\mathcal{C}^{\perp})_{F_q} \tag{16.1}$$

Proof: Let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{F}_q^n$. Then $\alpha \in tr(\mathcal{C})^\perp$ if and only if $\sum_{i=1}^n \alpha_i tr(x_i) = 0$ for all $x = (x_1, x_2, \ldots, x_n) \in \mathcal{C}$. Because of the linearity of the trace this is equivalent with $tr(\sum_{i=1}^n \alpha_i x_i) = 0$. We claim that this is equivalent with $\sum_{i=1}^n \alpha_i x_i = 0$. In fact, if this was not the case, then because of the F-linearity of $\mathcal{C}$ it would follow that tr is the 0-mapping, contradiction. We conclude that $\alpha \in tr(\mathcal{C})^\perp$ if and only if $\alpha \in \mathcal{C}^\perp$ and all $\alpha_i \in \mathbb{F}_q$. This is our claim. ∎

Because of the duality between strength and minimum distance this gives us the opportunity to construct linear q-ary codes as subfield codes $(\mathcal{C}^\perp)_{F_q}$. Knowledge of the minimum distance of this code is equivalent with knowledge of the strength of the corresponding tracecode $tr(\mathcal{C})$. In this chapter we will apply this construction to codes $\mathcal{C}$, whose codewords are given as evaluations of all polynomials of degree $< t$ for some t. The determination of the strength of the trace code will be an easy consequence of Lagrange interpolation.

Theorem (16.2) (Lagrange interpolation) *Let F be a field. A nonzero polynomial $p(X) \in F[X]$ of degree n has at most n roots.*
If n different elements $x_1, x_2, \ldots, x_n$ in F and n arbitrary elements $\alpha_1, \ldots, \alpha_n$ in F are given, then there is exactly one polynomial $p(X)$ of degree $< n$ with coefficients in F, which satisfies $p(x_i) = \alpha_i, i = 1, 2, \ldots, n$.

Proof: If $p(x) = 0$, then $p(X)$ is divisible by the linear factor $X - x$. If $p(X)$ has $n + 1$ different roots $x_1, x_2, \ldots, x_{n+1}$, then $p(X)$ is divisible by the product of the corresponding linear factors. If the degree of $p(X)$ is $\leq n$, it follows that $p(X)$ is the 0-polynomial. This proves the first claim. Consider the second claim. We prove at first the uniqueness of the polynomial. In fact, if $p_1(X)$ and $p_2(X)$ are polynomials both satisfying our conditions, then $p_1(X) - p_2(X)$ has n different roots, hence is the 0-polynomial, by the first claim. The existence of an interpolating polynomial it proved by writing it down:

$$p(X) = \sum_{i=1}^n \alpha_i \frac{\prod_{j \neq i}(X - x_j)}{\prod_{j \neq i}(x_i - x_j)}.$$

∎

Another way of looking at Lagrange interpolation is given by the Vandermonde determinant. Taking the powers of X as a basis of $\mathcal{RS}(k, q)$ we obtain a generator with columns indexed by the elements $u \in \mathbb{F}_q$, and with entry u^i in column u, row i, where $i = 0, 1, 2, \ldots, n-1$. The restriction onto any n columns is a Vandermonde matrix. The fact that $\mathcal{RS}(n, q)$ has strength n follows therefore also from the fact that the determinant of the Vandermonde matrix is nonzero. The following slight generalization of Lagrange interpolation follows as a corollary:

Theorem (16.3) *Let $l \geq 0$. If n different elements $x_1, x_2, \ldots, x_n$ in F and n arbitrary elements $\alpha_1, \ldots, \alpha_n$ in F are given, then there is exactly one polynomial $p(X)$ of degree $< n$ with coefficients in F, $x_i^l p(x_i) = \alpha_i, i = 1, 2, \ldots, n$.*

We aim at another basic fact concerning the relation between a code defined over a large field and its subfield and trace codes.

Definition (16.2) *Let $F = \mathbb{F}_{q^r} | \mathbb{F}_q$ and let $\mathcal{C}$ be an F-linear code. We call $\mathcal{C}$* **Galois invariant** *if $\mathcal{C} = \mathcal{C}^q$. This means that $\mathcal{C}$ is mapped onto itself by the Frobenius automorphism (see Lemma (2.1)).*

Theorem (16.4) *Let $\mathcal{C}$ be linear code over $F = \mathbb{F}_{q^r}$. Let $tr : F \longrightarrow \mathbb{F}_q$ be the trace. Assume $\mathcal{C} = \mathcal{C}^q$. Then the following hold:*

- *$tr(\mathcal{C}) = \mathcal{C}|_{F_q}$.*
- *The $\mathbb{F}_q$-dimension of $tr(\mathcal{C})$ equals the F-dimension of $\mathcal{C}$.*

Proof: The definition of the trace shows that $tr(\mathcal{C}) \subseteq \mathcal{C}|_{F_q}$. Elements of the subfield code, which are linearly independent over $\mathbb{F}_q$, are also linearly independent over F. This shows $dim_{F_q}(\mathcal{C}|_{F_q}) \leq dim_F(\mathcal{C})$. The last link in the chain is provided by Theorem (16.1). If the length of $\mathcal{C}$ is n, then

$$dim(tr(\mathcal{C})) = dim((C^{\perp}|_{F_q})^{\perp}) = n - dim(\mathcal{C}^{\perp}|_{F_q}) \geq n - dim_F(\mathcal{C}^{\perp}) = dim_F(\mathcal{C}).$$

It follows that we have equality all along.∎

16.2 Cyclic codes

Definition (16.3) *Let $n \mid q^r - 1$,*
$A \subseteq \mathbb{Z}/n\mathbb{Z}, F = \mathbb{F}_{q^r}, W$ the multiplicative subgroup of order n of F. Denote by $\mathcal{P}(A)$ the space of polynomials with coefficients in F, whose monomials have degrees in A.
Consider the array $\mathcal{B}(A,n,q) = \mathcal{B}(A)$, whose rows are indexed by $p(X) \in \mathcal{P}(A)$, whose columns are indexed by the elements $u \in W$, and corresponding entry $p(u)$. The **cyclic code** *$\mathcal{C}(A,n,q) = \mathcal{C}(A)$ is*

$$\mathcal{C}(A) = tr(\mathcal{B}(A))^{\perp}.$$

Call A the **defining set** *of $\mathcal{C}(A)$.*

Observe that we limit ourselves to the case when n is coprime to q. If we start from such a number n, then r is determined as the smallest number such that n divides $q^r - 1$. Equivalently we may describe r as the order of q mod n. We will show how to determine the dimension of cyclic codes and how to obtain lower bounds on the minimum distance. In order to determine the minimum distance we need the notion of cyclotomic cosets:

Definition (16.4) (cyclotomic cosets) *Consider the Galois group $G(\mathbb{F}_{q^r}|\mathbb{F}_q) =<\sigma>$ as in Definition (2.2), where σ is the Frobenius automorphism, let $n \mid q^r - 1$. Then G acts as a permutation group on $\{1,2,\dots,n\}$ in the following natural way:*

$$\sigma(i) = \lfloor iq \rfloor,$$

where $\lfloor j \rfloor$ denoted the representative of the integer j mod q^r-1 in $\{1,2,\dots,n\}$. The **cyclotomic cosets** *are the orbits of G. Denote by $Z_n(t)$ the orbit containing t.*

Definition (16.5) *In the situation of Definition (16.4) let $A \subseteq \{1,2,\dots,n\}$ be a subset. The* **Galois closure** *$\tilde{A}$ is defined as the union of the cyclotomic cosets intersecting A nontrivially.*

Let $\mathcal{B}(A,n,q) = \mathcal{B}(A)$ as in Definition (16.3). By definition the codes $\mathcal{B}(A)$ and $\mathcal{B}(\tilde{A})$ have the same trace-code. As $\mathcal{B}(\tilde{A})$ is Galois invariant, we can use Theorem (16.4) and conclude that the $\mathbb{F}_q-$dimension of $tr(\mathcal{B}(A))$ agrees with the F-dimension of $\mathcal{B}(\tilde{A})$. This dimension clearly is $|\tilde{A}|$. As $\mathcal{C}(A)$ is the dual of $tr(\mathcal{B}(A))$ we have proved the following:

Theorem (16.5) *The dimension of the code $\mathcal{C}(A)$ as given in Definition (16.3) is the sum of the lengths of the cyclotomic cosets which intersect A in the empty set.*

We turn to the minimum distance. Recall that our codes are unchanged if the defining set is replaced by its Galois closure. If we wish we can assume $A = \tilde{A}$. If A contains a set of t consecutive numbers, then it follows from Theorem (16.3) that $tr(\mathcal{B}(A))$ has strength $\geq t$. It follows that $\mathcal{C}(A)$ has minimum distance $\geq t+1$. We formalize this argument and generalize it slightly:

Definition (16.6) (intervals) *Let A be a subset of the cyclic group of order n : $A \subseteq \mathbb{Z}/n\mathbb{Z}$. Then A is an* **interval** *if there is a generator of $\mathbb{Z}/n\mathbb{Z}$ (equivalently: an integer j, coprime with n), such that $A = \{jl, j(l+1), \ldots, j(l+i-1)\}$, for some l (mod n).*

As we are free to replace the generator of our cyclic group by some other generator it is clear that our argument is valid in this situation. We have proved the following:

Theorem (16.6) (BCH-bound) *If $\tilde{A}$ contains an interval of length t, then $d(\mathcal{C}(A,n,q)) \geq t+1$.*

This is known as the BCH-bound. The reader should note the degree of liberty arising from the choice of the generator. This may not be intuitively clear from the notion of an interval. As an example, let $n = 35, j = 12$. Then $\{0, 12, 24, 36, 48\} = \{0, 1, 12, 13, 24\}$ is an interval of length 5.

If we wish to guarantee a certain minimum distance then it suffices by the BCH-bound to choose as defining set an interval A of a certain size.

Definition (16.7) (BCH-codes) *The BCH-code $\mathcal{BCH}(l,t,n,q)$ is the q-ary cyclic code of length n, where $n \mid q^r - 1$, with defining set $\mathcal{A} = \{l, l+1, \ldots, l+t-1\} = [l, l+t-1]$.*

In case $l = 1$ one speaks of BCH-codes in the narrow sense. A BCH-code is primitive if $n = q^r - 1$. The dimension of $\mathcal{BCH}(l,t,n,q)$ is $n - |\tilde{A}|$ (see Theorem (16.5)). Its minimum distance is $\geq t+1$ by the BCH-bound.

$t+1$ is customarily called the **designed distance.** This is in fact not quite appropriate. The designed distance ought better be defined as one more than the size of the largest interval contained in $\tilde{A}$, as this is the best lower bound obtainable by our present methods.

16.2.1 BCH-codes and their standard lengthenings

Let $n \mid q^r-1$ and $A=\{l,l+1,\dots,l+t-1\}$. The cyclic code $\mathcal{C}=\mathcal{C}([l,l+t-1])$ with defining set A is a BCH-code of dimension $k=n-\mid \tilde{A}\mid$ and designed distance $\geq t+1$. If we vary l and t we obtain a two-parameter family of BCH-codes. This situation is ideally suited for lengthening methods like construction X and construction XX (see Theorems (10.2) and (10.3)). We want to apply this technique in an important parametric situation.
Consider code $\mathcal{C}$ above and its subcodes $\mathcal{C}_l=\mathcal{C}([l-1,l+t-1]),\mathcal{C}_r=\mathcal{C}([l,l+t])$. Then $\mathcal{C}_l,\mathcal{C}_r$ have designed distance $\geq t+2$. It follows right from the definition of cyclic codes that $\mathcal{C}_l\cap\mathcal{C}_r=\mathcal{C}([l-1,l+t])$, another BCH-code, with designed distance $\geq t+3$. Moreover we assume A has been chosen such that $l-1$ and $l+t$ are not contained in $\tilde{A}$. We know that $dim(\mathcal{C}/\mathcal{C}_l)=\mid Z_n(l-1)\mid$ and $dim(\mathcal{C}/\mathcal{C}_r)=\mid Z_n(l+t)\mid$. We can apply Theorem (10.2) to one of the chains $\mathcal{C}\supset\mathcal{C}_l$ or $\mathcal{C}\supset\mathcal{C}_r$. Let us try to apply Theorem (10.3). We distinguish two cases: in the generic case when $l-1$ and $l+t$ are in different cyclotomic cosets, we can apply Theorem (10.3) to the codes $\mathcal{C},\mathcal{C}_l,\mathcal{C}_r,\mathcal{C}_l\cap\mathcal{C}_r$ and obtain a code of dimension k, minimum distance $\geq t+3$. In the case when $l-1$ and $l+t$ are in the same cyclotomic coset we have $\mathcal{C}_l=\mathcal{C}_r$. We can then apply Theorem (10.2) to the chain $\mathcal{C}\supset\mathcal{C}_l$, using the all even-code $[\mid Z_n(l-1)\mid+1,\mid Z_n(l-1)\mid,2]$ as auxiliary code. We collect our results in the following Theorem:

Theorem (16.7) (standard lengthening) *Let $n \mid q^r-1, A=\{l,l+1,\dots,l+t-1\}$ such that $l-1$ and $l+t$ are not contained in $\tilde{A}$. Let $\mathcal{C}$ be the cyclic code with A as defining set. Denote by $k=n-\mid\tilde{A}\mid$ the dimension of $\mathcal{C}$.*

Then the following hold:

- *Application of Theorem (10.2) yields codes*

$$[n+\mid Z_n(l-1)\mid,k,t+2]$$

and

$$[n+ \mid Z_n(l+t) \mid, k, t+2].$$

- *Assume $l-1$ and $l+t$ are not in the same cyclotomic coset. Then a code*
$$[n+ \mid Z_n(l-1) \mid + \mid Z_n(l+t) \mid, k, t+3]$$
can be constructed by applying Theorem (10.3) to $\mathcal{C}$.

- *Assume $l-1$ and $l+t$ are in the same cyclotomic coset. Then a code*
$$[n+ \mid Z_n(l-1) \mid +1, k, t+3]$$
can be constructed by applying Theorem (10.2) to $\mathcal{C}$.

Observe that we can apply the Theorem also to subcodes of $\mathcal{C}$. In the following section we give lists of examples of BCH-codes and their standard lengthenings.

16.3 Examples of BCH-codes and standard lengthenings

In the following subsections we list the cyclotomic cosets, the defining intervals $A = \{l, l+1, \ldots, l+t-1\}$, abbreviated to $[l, l+t-1]$, the parameters of the corresponding BCH-code and the code obtained by application of Theorem (16.7).

16.3.1 q=2,r=3,n=7

The cyclotomic cosets are $\{0\}, \{1,2,4\}, \{3,6,5\}$.

A	BCH-code	lengthening
$\emptyset$	[7,7,1]	[11,7,3]
[1,2]	[7,4,3]	[11,4,5]

Code $\mathcal{C}([1,2])$ is the familiar binary Hamming code.

16.3.2 q=2,r=4,n=15

cyclotomic cosets
0
1,2,4,8
3,6,12,9
5,10
7,14,13,11

A	BCH-code	lengthening
$\emptyset$	[15,15,1]	[20,15,3]
[1,2]	[15,11,3]	[20,11,5]
[1,4]	[15,7,5]	[18,7,7]
[1,6]	[15,5,7]	[20,5,9]

Observe that a binary code $[19, 7, 8]$ can also be obtained via shortening from the binary Golay code (see Chapter 11).

16.3.3 q=2,r=5,n=31

cyclotomic cosets
0
1,2,4,8,16
3,6,12,24,17
5,10,20,9,18
7,14,28,25,19
11,22,13,26,21
15,30,29,27,23

A	BCH-code	lengthening
[1,2]	[31,26,3]	[37,26,5]
[1,4]	[31,21,5]	[37,21,7]
[1,6]	[31,16,7]	[37,16,9],[41,16,11]
[1,10]	[31,11,11]	[37,11,13],[41,11,15],[39,10,15]
[1,14]	[31,6,15]	[37,6,17],[41,6,19],[47,6,23],[52,6,25]
[1,14]	[31,6,15]	[52,6,25],[43,5,21]
[1,31]	[31,1,31]	

The auxiliary codes used in the applications of Theorem (10.2) are $[10,5,4],[16,5,8],[21,5,10]$ (from Subsection 16.3.2), the extended Hamming code $[8,4,4]$ and $[12,4,6]$ (from Subsection 16.3.1, after addition of a parity check symbol).

16.3.4 q=3,r=3,n=26

The interval $A=[1,12]$ is disjoint from the cyclotomic cosets $\{0\},\{13\},\{14,16,22\}$ and $\{17,23,25\}$. We obtain a code $[28,8,15]_3$.

16.4 Ovoid codes

Let $r=2, n=q^2-1$. Consider the interval $A=\{1\ldots,q^2-q-3\}$. Observe that q^2-q-1, q^2-2 is a cyclotomic coset of length 2 avoiding A. As 0 and q^2-q-2 form cyclotomic cosets of length one each, we can apply Theorem (16.7) and obtain the **ovoid code** $[q^2+1,4,q^2-q]_q$.

16.5 Remarks

- The book [15] by Mc Williams and Sloane is the bible of traditional coding theory. Van Lint's book [14] is a very popular introduction to the subject.
- The traditional theory of cyclic codes uses a different approach, where cyclic codes are viewed as ideals in a certain factor ring of the ring of polynomial functions. We believe that the approach used here is more direct and more elegant.
- The most powerful method of code construction was introduced by Goppa and Manin. It is based on the theory of algebraic curves. The Reed-Solomon codes can be considered as the most elementary case of this construction, corresponding to **genus zero.** A very recommendable introduction to this method is Stichtenoth's book [22]. It is self-contained in that it also presents an introduction to the theory of algebraic curves.

- The data base [9] contains the best currently known upper and lower bounds on the maximum minimum distance of binary, ternary and quaternary linear codes of moderate length.

16.6 Problems

1. Lagrange interpolation: Determine the polynomial $p(X)$ of degree at most 2, which satisfies $p(0) = -1.5, p(-0.5) = 2$ and $p(2) = 1$.

2. Use Lagrange interpolation to show that the Reed-Solomon code $\mathcal{RS}(k, q)$ is an orthogonal array of strength k.

3. Show that the primitive BCH-code $\mathcal{C}([1, q^r - 2], q^r - 1, q)$ has dimension 1 and is in fact the repetetion code.

4. Show that the binary cyclic code of length 7, with defining set $A = \{1\}$ has parameters $[7, 4, 3]$ and is the binary Hamming code as mentioned first in the Introduction.

5. Construct a quaternary cyclic code $[17, 4, 12]_4$ (an **ovoid code**).

6. Construct a narrow-sense ternary BCH-code $[56, 10, 28]$ in case $r = 6, n = 56$ and show that it can be lengthened to yield a code $[58, 10, 30]_3$.

Chapter 17

Encoding and decoding cyclic codes

17.1 Encoding cyclic codes

We consider a q-ary cyclic code $\mathcal{C}(A)$ of length n, where n divides $q^r - 1$. The terminology is as introduced in Section 16.2.
Encoding is equivalent to a description of a generator matrix of $\mathcal{C}(A)$. We know that $k = dim(\mathcal{C}(A)) = n - \mid \tilde{A} \mid$, where $\tilde{A}$ is the union of all cyclotomic cosets intersecting A nontrivially (Theorem (16.5)). Here $\tilde{A}$ is the Galois closure of A, see Definition (16.5) Equivalently k may be described as the sum of the cardinalities of all cyclotomic cosets intersecting A trivially. This suggests, that every cyclotomic coset Z satisfying $Z \cap \tilde{A} = \emptyset$ should contribute $\mid Z \mid$ rows to a generator matrix of $\mathcal{C}(A)$. The following Theorem makes this precise. The fundamental Theorem (16.4) is instrumental again.

Theorem (17.1) *Let $\mathcal{C}(A) = \mathcal{C}(A, n, q)$ be a q-ary cyclic code of length n. Use the standard terminology which we recalled in the beginning of this chapter. Let $F = I\!F_{q^r} = I\!F_q(\epsilon)$. Let further $Z = Z(i)$ be a cyclotomic coset satisfying $Z \cap \tilde{A} = \emptyset$. Put $\mid Z \mid = s$. Define $v(j, -i) \in I\!F_q^n, j = 0, 1, \ldots, r-1$ as the n-tuple with entry $tr(\epsilon^j \beta^{-i})$ in coordinate $\beta \in W$. We can choose a set $S(-i)$ of s values of j such that the matrix with rows $v(j, -i), j \in S(-i)$ has rank s. Define a matrix $G(A)$ with the vectors $v(j, -i)$ as its rows, where i varies of representatives of cyclotomic cosets avoiding $\tilde{A}$, and $j \in S(-i)$.
$G(A)$ is a generator matrix of $\mathcal{C}(A)$.*

Proof: Theorem (16.5) shows that the number of rows of $G(A)$ equals the dimension k of $\mathcal{C}(A)$. Also $v(j,-i)$ belongs by definition to $tr(\mathcal{B}(-Z,n,q))$. As $-Z$ is a cyclotomic coset it follows that the trace-code of $\mathcal{B}(-Z,n,q)$ equals the subfield code (see Theorem (16.4)). This shows that

$$v(j,-i) \in tr(\mathcal{B}(-Z)^{\perp}) = tr(\mathcal{B}([0,n-1] \setminus Z)^{\perp}) \subseteq tr(\mathcal{B}(A)^{\perp}) = \mathcal{C}(A).$$

We are done.■

17.1.1 A binary example

We consider the primitive binary BCH-code $\mathcal{C}(A)$ of length 15 and designed distance 7. This means $A = [1,6]$. The field $F = \mathbb{F}_{16}$ has been described in Chapter 2. We leave it as an easy exercise to determine the cyclotomic cosets. Our code has dimension 5 and minimum distance ≥ 7. We have $[1,15] \setminus \tilde{A} = \{0,7,11,13,14\} = Z(0) \cup Z(7)$. As $-Z(0) = Z(0)$ and $-Z(7) = Z(1)$ we can choose $-i = 0$ and $-i = 1$. This leads to the following generator matrix $G(A)$:

index:	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$v(3,0)$:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$v(0,1)$:	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0
$v(1,1)$:	0	1	1	1	1	0	1	0	1	1	0	0	1	0	0
$v(2,1)$:	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0
$v(3,1)$:	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1

Let us see we understand how this matrix was calculated. The column indexed t corresponds to α^t. Here α is a primitive element of F. The first row is indexed $v(3,0)$. This corresponds to the monomial α^3 of degree 0. All entries in this row are $tr(\alpha^3) = 1$. This explains why we chose $j = 3$. Any other choice of j would have given us a 0-row. In the language of Theorem (17.1) we have $S(0) = \{3\}$. Rows $v(j,1), j = 0,1,2,3$ correspond to the monomials $\alpha^j X$. Consider the column of $G(A)$, which is indexed by 6, say. The corresponding entries are $tr(\alpha^j \alpha^6), j = 0,1,2,3$, concretely $tr(\alpha^6) = 1, tr(\alpha^7) = 1, tr(\alpha^8) = 0$ and $tr(\alpha^9) = 1$. This explains the remaining entries in column 6.

17.2 The Euclidean algorithm

As everybody knows the Euclidean algorithm is a fast method to compute the **greatest common divisor** (a, b) of two elements a, b of a Euclidean domain. The Euclidean domain R comes with a norm N and the basis of the Euclidean algorithm is division with remainder:

$$a = q \cdot b + r,$$

where either $r = 0$ or $N(r) < N(b)$. consider the domain $\mathbb{Z}$ with norm $N(x) = |x|$. As an example we determine $(910, 143)$:

910	=	6	·	143	+ 52
143	=	2	·	52	+ 39
52	=	1	·	39	+ 13
39	=	3	·	13	+ 0

As soon as the remainder becomes 0 we read off the gcd, in the example

$$(910, 143) = (143, 52) = (52, 39) = (39, 13) = (13, 0) = 13.$$

The first columns contains the remainers: $r_{-1} = a = 910, r_0 = b = 143, r_1 = 52, r_2 = 39$, the second column contains the quotients: $q_{-1} = 6, q_0 = 2, q_1 = 1, q_2 = 3$. Row k is then: $r_k = q_k \cdot r_{k-1} + r_{k-2}$.

Another consequence, both of practical and theoretical importance, is the fact that (a, b) can be written as a linear combination of a and b :

$$(a, b) = u \cdot a + v \cdot b.$$

In our example we start from the penultimate row. It has the gcd as remainder. We solve for the remainder and obtain $13 = 1 \cdot 52 - 1 \cdot 39$, an expression of 13 as linear combination of 52 and 39. Then we work our way up. The previous row expresses 39 as a linear combination of 143 and 52. Substituting this into our expression we obtain $13 = 3 \cdot 52 - 1 \cdot 143$. Proceeding like this once more we obtain $13 = 3 \cdot 910 - 19 \cdot 143$. This is the desired expression $(910, 143) = u \cdot 910 + v \cdot 143$. Here $u = 3, v = -19$.

We want to generate these data by the algorithm, without having to go back. That is why we introduce two more columns, with entries u_k, v_k such that $r_k = u_k \cdot a + v_k \cdot b$. The initialization is easy. The first two rows with indices -1 and 0 are

k	Q	R	U	V
-1		a	1	0
0		b	0	1

Row $k+1$ will be calculated from the two preceding rows. So assume rows $k-1$ and k are given, as follows:

Q	R	U	V
q_{k-1}	r_{k-1}	u_{k-1}	v_{k-1}
q_k	r_k	u_k	v_k

We apply long division to the two preceding remainders: $r_{k-1} = q_{k+1} \cdot r_k + r_{k+1}$. This gives us the new remainder r_{k+1} and the new quotient q_{k+1}. Moreover we have $r_k = u_k \cdot a + v_k \cdot b$ and $r_{k-1} = u_{k-1} \cdot a + v_{k-1} \cdot b$. This yields $r_{k+1} = r_{k-1} - q_{k+1} \cdot r_k = (u_{k-1} - q_{k+1} u_k) \cdot a + (v_{k-1} - q_{k+1} v_k) \cdot b$. We obtain the recursions

$$u_{k+1} = u_{k-1} - q_{k+1} u_k \text{ and } v_{k+1} = v_{k-1} - q_{k+1} v_k.$$

Here is the scheme for our example:

Q	R	U	V
	910	1	0
	143	0	1
6	52	1	-6
2	39	-2	13
1	13	3	-19
3	0	-11	70

As soon as 0 appears in column R, we know that the gcd is in the preceding row. There is more information in the complete scheme:

Theorem (17.2) (Cross-product theorem) *In the complete Euclidean scheme for the calculation of the greatest common divisor (a, b) the following relations hold:*

- $r_{k-1} u_k - r_k u_{k-1} = \pm b$
- $r_{k-1} v_k - r_k v_{k-1} = \pm a$

- $u_{k-1}v_k - u_kv_{k-1} = \pm 1$

Proof: These elementary statements are proved by induction. We consider only the first case: $r_ku_{k+1} - r_{k+1}u_k = r_k(u_{k-1} - q_{k+1}u_k) - (r_{k-1} - q_{k+1}r_k)u_k = r_ku_{k-1} - r_{k-1}u_k$, which is the negative of the cross-product one step earlier.It suffices to check the cross-product for the two initial rows, and this is $-b$.
The third statement of Theorem (17.2) shows that $(u_k, v_k) = 1$ for all k. Consider the last row (row j say) of the complete schem, put $u = u_j, v = v_j$. We have $r_j = 0 = u \cdot a + v \cdot b$. It follows $v/u = -a/b$. As $(u, v) = 1$ we have that v/u is the reduced form of $-a/b$.

Lemma (17.1) *The norms of the elements in the R-column are strictly decreasing starting from row 0.*
The norms of the elements in the U-column and in the V-column are strictly increasing starting from row 1.

Proof: The statement concerning row R is obvious from the way the algorithm was defined (division with remainder). Consider the recursion $u_{k+1} = u_{k-1} - q_{k+1}u_k$. Proceed by induction. Assume $N(u_{k-1}) \leq N(u_k)$. As $N(q_{k+1}) > 0$, it follows that $N(u_{k+1}) = N(q_{k+1}u_k) > N(u_k)$. The same argument holds for the V-column.■

17.3 Decoding BCH-codes

We consider the q-ary BCH-code (equivalently: cyclic code) of length n, where n divides $q^r - 1$, with designed distance $2e + 1$ (although the term *designed distance* is not really appropriate),with defining set $A = \{l, l + 1, \ldots, l + e - 1\} = [l, l + 2e - 1]$. Decoding is based on a check matrix for $\mathcal{C}(A)$.

Definition (17.1) *H is a $(2e, n)$-matrix with rows indexed by the elements of A (the natural numbers from l to $l + 2e - 1$), columns indexed by the elements $\beta \in W$ and entry $\beta^i \in F$ in row i, column β.*

Clearly H is a generator matrix of the array $\mathcal{B}(A)$ in the sense that the rows of $\mathcal{B}(A)$ are the F-linear combinations of the rows of H. In particular an n-tuple $c = (c_\beta) \in \mathbb{F}_q^n$ is contained in $\mathcal{C}(A)$ if and only if its dot products with the rows of $tr(H)$ vanish. This is equivalent with the vanishing of the F-linear dot products between c and the rows of H. These products are known as the syndromes.

Definition (17.2) (syndrome) *Let $i \in [l, l+2e-1] = A$ and $c = (c_\beta), \beta \in W$. The syndrome $S_i \in F$ is defined as*

$$S_i = S_i(c) = \sum_{\beta \in W} c_\beta \cdot \beta^i.$$

The **syndrome polynomial** *is $s(z) = \sum_{i=0}^{2e-1} S_{i+l} z^i \in F[z]$.*

We see that S_i is the dot product between c and row i of the check matrix H. In particular we have $c \in \mathcal{C}(A)$ if and only $S_1 = S_2 = \ldots = S_{2e} = 0$.

17.3.1 The fundamental equation

Let $x = (x_\beta) \in \mathcal{C}(A)$ be the code-word, which was sent. The received word is $y = (y_\beta) \in \mathbb{F}_q^n$. Define the error-vector as $err = y - x = (e_\beta)$. We have $S_i(err) = S_i(y)$ for all i. We work under the assumption that no more than e errors occurred, equivalently that the vector err has weight $\leq e$. Let $M = \{\beta \mid e_\beta \neq 0\}$ be the set of coordinates, where errors occurred. We express the syndrome polynomial in a different way, based on the error-vector:

$$s(z) = \sum_{i=0}^{2e-1} S_{i+l} z^i = \sum_{i=0}^{2e-1} \sum_{\beta \in M} e_\beta u^{i+l} z^i = \sum_{\beta \in M} e_\beta \beta^l \sum_{i=0}^{2e-1} (\beta z)^i.$$

We use the geometric series and get

$$s(z) = \sum_{\beta \in M} e_\beta \beta^l (1 - (\beta z)^{2e})/(1 - \beta z) =$$

$$\sum_{\beta \in M} e_\beta \beta^l/(1 - \beta z) - \sum_{\beta \in M} e_\beta \beta^{2e+l} z^{2e}/(1 - \beta z).$$

The common denominator for these rational functions is

$$l(z) = \prod_{\beta \in M} (1 - \beta z).$$

The polynomial $l(z)$ is known as the **error locator.** The roots of $l(z)$ are precisely the inverses of the coordinates, where errors have occurred. With $l(z)$ as denominator we have

$$s(z) = \frac{w(z)}{l(z)} - \frac{u(z)}{l(z)} z^{2e}$$

and it is clear how the polynomials $w(z), u(z)$ look like. After multiplication with $l(z)$ we obtain the fundamental equation:

Theorem (17.3) (fundamental equation) *We have*

$$s(z) \cdot l(z) = w(z) - u(z) \cdot z^{2e}.$$

Here $l(z)$ is the error locator as introduced above and we have

$$w(z) = \sum_{\beta \in M} e_\beta \beta^l \prod_{\gamma \in M, \gamma \neq \beta} (1 - \gamma z),$$

$$u(z) = \sum_{\beta \in M} e_\beta \beta^{2e+l} \prod_{\gamma \in M, \gamma \neq \beta} (1 - \gamma z).$$

Polynomial $w(z)$ is known as the **error evaluator.**

Observe that the receiver can calculate $s(z)$ and needs to know $l(z)$ and $w(z)$. In the binary case $q = 2$ it suffices to know $l(z)$.

17.3.2 Using Euclid's algorithm

The idea, which leads to the desired decoding algorithm, is based on the Euclidean algorithm and goes as follows: Put $a = z^{2e}$ and $b = s(z)$. Apply the Euclidean algorithm to a and b. We work in the Euclidean domain $F[z]$ (a polynomial ring over a field), with the polynomial degree as Euclidean norm. Use the complete scheme of Section 17.2. We are not interested in the final result, the gcd of a and b. Rather we stop the algorithm at row j, which is defined by $deg(r_j) < e, deg(r_{j-1}) \geq e$. The basic observation is that $v_j(z)$ (in the terminology of Section 17.2) is up to a constant factor identical with $l(z)$. Once this is proved it is clear how to determine the set M of coordinates, which are in error: determine the roots of $v_j(z)$. The inverses of these roots are the coordinates β, where $y_\beta \neq x_\beta$. In the binary case $q = 2$ this suffices to decode.

Theorem (17.4) (the algorithm) *In the current terminology the following holds: apply the Euclidean algorithm to* $a = z^{2e}$ *and* $b = s(z)$ *in the form of the complete scheme. Determine row* j *by* $deg(r_j) < e, deg(r_{j-1}) \geq e$. *Assume not more than* e *errors have occurred. Then there is a constant* $\lambda = v_j(0) \in \mathbb{F}_q^*$ *such that*

$$\boxed{\begin{array}{rcl} v_j(z) & = & \lambda \cdot l(z) \\ r_j(z) & = & \lambda \cdot w(z) \\ u_j(z) & = & \lambda \cdot u(z) \end{array}}$$

The proof of Theorem (17.4) demands nothing but a good understanding of the Euclidean algorithm. Here is an easy Lemma, which will be needed:

Lemma (17.2) $(l(z), u(z)) = 1$.

Proof: As $l(z)$ is a product of linear factors we have to show that roots of $l(z)$ cannot be roots of $u(z)$. Let $\beta \in M$. Then $u(1/\beta) = e_\beta \beta^{2e+l} \prod_{\gamma \in M, \gamma \neq \beta}(1 - \gamma/\beta) \neq 0$.■

Consider the last rows before the algorithm stops:

q_{j-1}	$r_{j-1}(z)$	$u_{j-1}(z)$	$v_{j-1}(z)$
q_j	$r_j(z) = w^*(z)$	$u_j(z) = u^*(z)$	$v_j(z) = l^*(z)$

We compare the starred polynomials with their non-starred counterparts. The defining property of the entries in columns U and V shows

$$w^*(z) = u^*(z)z^{2e} + l^*(z)s(z).$$

This means that the starred polynomials satisfy the fundamental equation of Theorem (17.3).

Lemma (17.3) *We have*

$$deg(l^*(z)) \leq e, deg(u^*(z)) < e \text{ and } deg(w^*(z)) < e.$$

Proof: In the case of $w^*(z)$ this belongs to the stop-condition. Use Theorem (17.2), the cross-product theorem. One of its statements is

$$l^*(z)r_{j-1}(z) - v_{j-1}(z)w^*(z) = \pm z^{2e}.$$

The monotonicity of the degrees (Lemma (17.1)) shows that the first summand is dominating on the left side. This shows $deg(l^*(z)r_{j-1}(z)) = 2e$.

It follows from the choice of j that $deg(r_{j-1}(z)) \geq e$, hence $deg(l^*(z)) \leq e$. In the case of $u^*(z)$ we proceed in an analogous way, starting from another statement of Theorem (17.2):

$$u^*(z)r_{j-1}(z) - u_{j-1}(z)w^*(z) = \pm s(z).$$

We obtain $deg(u^*(z)r_{j-1}(z)) = deg(s(z)) < 2e$ and finally $deg(u^*(z)) < e$, as required.■

We start from the fundamental equations, for the original polynomials and for the starred polynomials:

$l(z)s(z) + u(z)z^{2e}$	$=$	$w(z)$
$l^*(z)s(z) + u^*(z)z^{2e}$	$=$	$w^*(z)$

Multiply the first equation by $l^*(z)$, the second by $l(z)$ and subtract. This yields

$$(l^*(z)u(z) - l(z)u^*(z))z^{2e} = l^*(z)w(z) - l(z)w^*(z).$$

By definition $u(z)$ and $w(z)$ have degrees $\mid M \mid -1 < e$. It follows from Lemma (17.3) that the right side of our equation has degree $< 2e$. This shows

$l^*(z)u(z)$	$=$	$l(z)u^*(z)$
$l^*(z)w(z)$	$=$	$l(z)w^*(z)$

Next we use Lemma (17.2). The Euclidean algorithm shows that there exist polynomials $f(z), g(z)$ such that $1 = f(z)l(z) + g(z)u(z)$. Multiply both sides by $l^*(z) : l^*(z) = f(z)l(z)l^*(z) + g(z)l^*(z)u(z)$. Use the first equation above to substitute $l^*(z)u(z)$. We obtain

$$l^*(z) = (f(z)l^*(z) - g(z)u^*(z))l(z).$$

We have found a polynomial $k(z)$ such that $l^*(z) = k(z)l(z)$. The equations above yield $w^*(z) = k(z)w(z)$ and $u^*(z) = k(z)u(z)$. As $u_j(z) = u^*(z)$ and $v_j(z) = l^*(z)$ and ± 1 can be written as a linear combination of $u_j(z)$ and $v_j(z)$ by Theorem (17.2), we have that $(u^*(z), v^*(z)) = 1$. This shows that the polynomial $k(z)$ is indeed a nonzero constant λ. As $l(0) = 1$ we have $\lambda = l^*(0)$. Comparison with the expressions in the above equations shows that the proof of Theorem (17.4) is complete. ■

17.3.3 The binary example continued

We consider the primitive binary BCH-code $\mathcal{C}(A)$ of length 15 and designed distance 7 of Subsection 17.1.1. We organize the checkmatrix H such that the columns are indexed by the powers of α, in descending order, the first column corresponding to α^{14}, the last to $\alpha^0 = 1$. The rows correspond to $X, X^2, \ldots, X^6$. The entries are powers of α again. We use the convention to write out the exponents only. The following matrix H is obtained. The first column is a numbering of the rows.

1	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2	13	11	9	7	5	3	1	14	12	10	8	6	4	2	0
3	12	9	6	3	0	12	9	6	3	0	12	9	6	3	0
4	11	7	3	14	10	6	2	13	9	5	1	12	8	4	0
5	10	5	0	10	5	0	10	5	0	10	5	0	10	5	0
6	9	3	12	6	0	9	3	12	6	0	9	3	12	6	0

The first row also serves to show the parametrization. Observe that the row of the generator matrix $G(A)$ as given in Section 17.1 are indeed orthogonal to the rows of H. It follows from the definition that $S_2 = S_1^2, S_4 = S_2^2 = S_1^4, S_6 = S_3^2$. In general we have $S_{2i} = S_i^2$. This shows that we do not need to write out all the rows of the check matrix but rather only rows forming a set of representatives of the cyclotomic cosets intersecting A nontrivially. In our cases it suffices to know rows number 1,3 and 5. Let

$$y = (1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1)$$

be the received vector. We calculate $S_1 = \alpha^{14} + \alpha^{10} + \alpha^9 + \alpha^8 + \alpha^5 + \alpha^2 + \alpha + 1 = 0$. It follows $S_2 = S_4 = 0$. Likewise $S_5 = 0$. However, we have $d \notin \mathcal{C}(A)$ as

$S_3 = \alpha^{12}+1+\alpha^{12}+\alpha^9+1+\alpha^6+\alpha^3+1 = \alpha^{12}$. The second nonzero syndrome is therefore $S_6 = S_3^2 = \alpha^9$. We obtain

$$s(z) = \alpha^9 z^5 + \alpha^{12} z^2.$$

We use the Euclidean algorithm:

k	Q	R	U	V
-1		z^6	1	0
0		$\alpha^9 z^5 + \alpha^{12} z^2$	0	1
1	$\alpha^6 z$	$\alpha^3 z^3$	1	$\alpha^6 z$
$j=2$	$\alpha^6 z^2$	$\alpha^{12} z^2$	$\alpha^6 z^2$	$\alpha^{12} z^3 + 1$

We see that we need only two divisions with remainder. We have to factor $l^*(z) = \alpha^{12} z^3 + 1$. Its roots correspond to $z^3 = \alpha^3$, and are therefore $\alpha, \alpha^6, \alpha^{11}$. The errors occurred in coordinates α^{14}, α^9 and α^4. Correcting d in these coordinates yields the vector

$$x = (0,0,0,0,1,0,1,0,0,1,1,0,1,1,1).$$

It is easy to see that this is indeed a code-word. In the notation of Section 17.1 we have $x = v(3,0) + v(0,1)$.

Let us see what happens if less than e errors occur. Let the received vector be

$$y = (0,0,0,0,1,0,1,1,0,1,1,0,1,1,1).$$

The syndrom vector is

$$s(z) = \alpha^{12} z^5 + \alpha^5 z^4 + \alpha^{13} z^3 + \alpha^6 z^2 + \alpha^{14} z + \alpha^7.$$

The Euclidean algorithm terminates earlier:

k	Q	R	U	V
-1		z^6	1	0
0		$s(z)$	0	1
$j=1$	$\alpha^3 z + \alpha^{11}$	α^3	1	$\alpha^3 z + \alpha^{11}$

As $v_1(z)$ has degree 1 we see that only one error occurred. The root of $v_1(z)$ is α^8. It follows that the error occurred in coordinate $\alpha^{-8} = \alpha^7$. The vector x, which was sent, is

$$x = (0,0,0,0,1,0,1,0,0,1,1,0,1,1,1),$$

the same word as before.

17.4 Remarks

The development of the theory is essentially in [19]. Pretzel also makes the following historical remarks: The first version of this algorithm is due to Berlekamp [4]. Massey modified it in [16] and showed a connection with shift-register sequences. Pretzel's presentation is based on a version introduced by Sugiyama et.al. [25], which uses the Euclidean algorithm. All these decoding methods, including another one based on continuous fractions, are essentially equivalent.

1. We give generator matrices and check matrices for cyclic codes, which are essentially canonical. $\mathcal{C}(A)$. Each cyclotomic coset Z intersection the Galois closure $\tilde{A}$ trivially contributes $\mid Z \mid$ rows to the generator matrix.
2. The **Euclidean algorithm** is reviewed and used to **decode BCH-codes.**

1. The **complete Euclidean algorithm** for the greatest common divisor (a, b).
2. a, b are from a Euclidean domain with norm function N.
3. **Initialization:** $r_{-1} = a, r_0 = b, u_{-1} = 1, u_0, v_{-1} = 0, v_0 = 1$.
4. The **step:** let $r_{k-1}, r_k, u_{k-1}, u_k = 0, v_{k-1}, v_k$ be given. **Long division** $r_{k-1} = q_{k+1} \cdot r_k + r_{k+1}$, where $N(r_{k+1}) < N(r_k)$ defines q_{k+1} and r_{k+1}.
5. $u_{k+1} = u_{k-1} - q_{k+1} u_k, v_{k+1} = v_{k-1} - q_{k+1} v_k$.
6. The **termination:** if k is minimal such that $r_{k+1} = 0$, then

$$(a, b) = r_k = u_k \cdot a + v_k \cdot b.$$

1. The **decoding algorithm for BCH-codes.**
2. The code is $\mathcal{C}(A)$, of length $n \mid q^r - 1$, with defining set $A = [l, l+2e-1]$ (of designed distance $2e+1$). The subgroup of order n of the multiplicative group of $F = \mathbb{F}_{q^r}$ is W.
3. The standard F-ary check matrix is H, with rows indexed by $i \in A$, columns indexed by $\beta \in W$, entries β^i.
4. Let x be the codeword sent, y the vector received, $err = y - x = (e_\beta)$ the error vector.
5. For $i \in A$ the **syndrome** is $S_i(y) = S_i(err) = \sum_{\beta \in W} y_\beta \beta^i \in F$, the dot product of y and row i of H.
6. The **syndrome polynomial** is $s(z) = \sum_{i=0}^{2e-1} S_{l+i} z^i \in F[z]$.
7. $M = \{\beta \in W \mid e_\beta \neq 0\}$ is the set of coordinates, where errors have occurred.
8. $l(z) = \prod_{\beta \in M}(1 - \beta z)$ is the **error locator:** its roots are the inverses of the error locations.
9. $w(z) = \sum_{\beta \in M} e_\beta \beta^l \prod_{\gamma \in M, \gamma \neq \beta}(1 - \gamma z)$ is the **error evaluator.**
10. **The algorithm:** apply the complete Euclidean algorithm to the polynomials $a = z^{2e}$ and $b = s(z)$, with the degree as norm function. Stop at the first index j where $deg(r_j) < e$.
11. $v_j(z) = \lambda \cdot l(z), r_j(z) = \lambda \cdot w(z)$, where $\lambda = v_j(0)$. This determines the location and values of the errors, provided no more than e errors have occurred.

Bibliography

[1] W.O.Alltop: *A method for extending binary linear codes, IEEE Trans. Inform. Theory* **30** (1984) 871-872.

[2] M.Atici, D.R.Stinson: *Universal hashing and multiple authentication,* manuscript.

[3] S.Ball, A.Blokhuis, F.Mazzocca: *Maximal arcs in desarguesian planes of odd order do not exist,* submitted for publication in *Combinatorica.*

[4] E.R. Berlekamp: *On decoding binary Bose-Chaudhuri-Hocquenghem codes, IEEE Transaction on Information Theory* **11** (1965),577-579.

[5] T.Beth,D.Jungnickel,H.Lenz: *Design Theory,* Bibliographisches Institut, Zürich 1985.

[6] C.H.Bennett,G.Brassard,J-M.Roberts: *Privacy amplification by public discussion,* SIAM Journal of Computation 17(1988),210-229.

[7] C.H.Bennett,G.Brassard,C. Crépeau, U.M.Maurer: *Generalized Privacy Amplification, IEEE Transactions on Information Theory* **41**(1995),1915-1923.

[8] J.Bierbrauer,T.Johansson,G.Kabatianskii,B.Smeets: *On families of hash functions via geometric codes and concatenation,* Advances in Cryptology - CRYPTO 93, Lecture Notes in Computer Science 773,331-342.

[9] A.E. Brouwer: Data base of bounds for the minimum distance for binary, ternary and quaternary codes,
URL http://www.win.tue.nl/win/math/dw/voorlincod.html or

URL http://www.cwi.nl/htbin/aeb/lincodbd/2/136/114 or
URL ftp://ftp.win.tue.nl/pub/math/codes/table[234].gz.

[10] K.A.Bush: *Orthogonal arrays of index unity,* Ann.Math.Stat. 23(1952),426-434.

[11] J.L.Carter,M.N.Wegman: *Universal classes of hash functions,* J. Computer and System Sci. 18(1979),143-154.

[12] S.C.Chang,J.K.Wolf: *A simple derivation of the Mac Williams' identity for linear codes, IEEE Transactions on Information Theory* **26**(1980),476-477.

[13] J.H.Griesmer: *A bound for error correcting codes,* IBM Journal Research Development 4(1960),532-542.

[14] J.H. van Lint: *Introduction to Coding Theory,* Springer 1982.

[15] F. J. MacWilliams, N. J. A. Sloane. *The Theory of Error-Correcting Codes,* North-Holland, 1977.

[16] J.L. Massey: *Shift register synthesis and BCH decoding, IEEE Transaction on Information Theory* **15** (1969),122-127.

[17] G.Pasquier: *The binary Golay code obtained from an extended cyclic code over* $\mathbb{F}_8$, *European Journal of Combinatorics* **1**(1980),369-370.

[18] V. Pless: *Decoding the Golay codes, IEEE Transactions on Information Theory* **32** (1986),561-567.

[19] O.Pretzel: *Error-correcting codes and finite fields,* Clarendon Press, Oxford 1992.

[20] C.E. Shannon: *A mathematical theory of communication,* Bell System Technical Journal 27 (1948),379-423,623-656.

[21] C.E. Shannon: *Communication Theory of Secrecy Systems,* Bell System Technical Journal 28 (1949),656-715.

[22] H.Stichtenoth: *Algebraic function fields and codes*, Springer 1993.

[23] D.R.Stinson: *Universal Hashing and Authentication Codes,* to appear in Designs, Codes and Cryptography.

[24] D.R.Stinson:*The combinatorics of authentication and secrecy codes,* Journal of Cryptology 2(1990),23-49.

[25] Y.Sugiyama,M.Kasahara,S.Hirasawa and T.Namekawa: *A method for solving key equation for decoding Goppa codes,* Information and Control **27**(1975),87-99.

[26] M.N.Wegman, J.L.Carter: *New hash functions and their use in authentication and set equality,* Journal of Computer and System Science 22 (1981),265-279.

[27] E. Witt: *Über Steinersche Systeme,* Abhandlungen Mathematisches Seminar Universität Hamburg **12** (1938),265-275.