AN INTRODUCTION TO ALGEBRAIC CODING THEORY

RUOPU HUANG

Abstract. Algebraic Coding Theory studies the design of error-correcting
codes for the reliable transmission of information across noisy channels. In
this paper, we discuss error-correcting codes. We begin with the fundamentals
of coding theory, and then explore linear codes, which are subspaces of vector
spaces. We then use results from abstract algebra to understand more complex
codes such as Hamming codes and Cyclic codes. We conclude by stating and
proving a fundamental result relating Hamming codes and Cyclic codes.

Contents

1. Introduction 1
2. Basic Definitions and Results 2
   2.1. Elementary Definitions and Examples 2
   2.2. Correcting and Detecting Errors 3
   2.3. Linear Codes 5
   2.4. Hamming code 7
3. Important Results From Algebra 8
   3.1. Abstract Algebra Review 8
   3.2. Cyclic Codes 10
Acknowledgments 12
References 12

1. Introduction

Algebraic Coding Theory is an area of discrete applied mathematics that is con-
cerned with developing error-control codes and encoding or decoding procedures.
Error-control codes are used to detect and correct errors that occur when data is
transmitted across some noisy channel. For example, CDs use error-control codes
so CD players can read data from a CD even if it has been corrupted by noise,
causing imperfections on the CD. Error-control codes build redundancy into a mes-
sage in the form of extra data. This allows the receiver to check consistency of the
delivered message, and recover data that has been corrupted.

An example of an error-control code is the ISBN code for books. An ISBN
codeword comprises 10 digits: $x_1x_2 \cdots x_{10}$ where $x_i \in \{0, 1, \cdots 9\}$. The first 9 digits
of each ISBN represent information about the book. The 10th digit is a redundancy,
or a check digit, chosen so the whole ten-digit string $x_1x_2 \cdots x_{10}$ satisfies:
\[ \sum_{i=1}^{10} ix_i \equiv 0 \pmod{11}. \]

It can be shown that the ISBN code can detect any single error.

The earliest known form of error-control codes came between 7th to 10th CE. A group of Jewish scribes created the Numerical Masorah to ensure accurate reproduction of the Hebrew Bible. This included counts of words in a line, section, book and groups of books. Standard became such that a deviation in even a single letter was considered unacceptable ([1]). The effectiveness of their error-controlling method was verified by the accuracy of copying through the centuries, demonstrated by discovery of the Dead Sea Scrolls in 1947-1956([2]).

Modern development of error-control codes features Claude Shannon, who published “A Mathematical Theory of Communication” in 1948. This work focuses on the problem of how best to encode the information a sender wants to transmit([3]). Then in 1949, the binary Golay code was developed. It is an error-correcting code capable of correcting up to 3 errors and detecting the 4th in a 24-bit word. In 1968, Richard Hamming won the Turing award for his work in error-correcting and error-detecting codes. He invented the concepts such as Hamming codes and Hamming distance([4]).

This paper develops the theory of Algebraic Coding Theory. In §2 we will introduce the definitions and results from coding theory and supplement this material with examples. In §3 the results from abstract algebra will then be reviewed, including the relation between Hamming codes and Cyclic codes.

2. Basic Definitions and Results

In §2.1, we will introduce definitions of terminologies in coding theory along with examples. In §2.2 we will discuss how errors can be detected and corrected depending on the properties of the codes. In §2.3 we will focus on linear codes, which are codes that possess the structure of a vector space. In §2.4 we study a particular class of linear codes known as Hamming codes. We will follow Spence’s exposition([5]) on algebraic coding theory throughout this section and the next.

2.1. Elementary Definitions and Examples.

**Definition 2.1.** Let \( F \) be a field and \( n \in \mathbb{N} \). A codeword is an element of \( F^n \). The field \( F \) is sometimes known as the alphabet. The length of any codeword \( x \in F^n \) is said to be \( n \). A code is a subset \( X \subseteq F^n \) such that for all \( x \in X \), \( x \) is a codeword. A code is \( q \)-ary if its codewords are defined over the \( q \)-ary alphabet \( F_q \). A code is binary if it is 2-ary.

**Example 2.2.** For the ISBN code, \( n = 10 \) and the alphabet is \( F_{11} \). We can show \( 0 - 13165 - 332 - 6 \) is a valid ISBN codeword by calculating the weighted check sum:

\[
2(1) + 3(3) + 4(1) + 5(6) + 6(5) + 7(3) + 8(3) + 9(2) + 10(6) = 198 \equiv 0 \mod 11.
\]

We can show that \( 0 - 13924 - 101 - 4 \) is not a valid ISBN codeword by calculating the weighted check sum:
2(1) + 3(3) + 4(9) + 5(2) + 6(4) + 7(1) + 9(1) + 10(4) = 137 \not\equiv 0 \mod 11.

**Definition 2.3.** Let \( x, y \) be codewords. The Hamming weight \( w(x) \) of \( x \) is the number of nonzero components in the codeword. The Hamming distance between two codewords \( d(x, y) \) is the Hamming weight of the vector difference \( x - y \), i.e \( w(x - y) \).

**Example 2.4.** Consider the code \( C = \{(111010), (100001)\} \subset \mathbb{F}_2^6 \). Then \( w(111010) = 4, w(100001) = 2 \), and \( d(111010, 100001) = 4 \).

**Definition 2.5.** The minimum (Hamming) distance of a code \( C \), denoted as \( d(C) \), is the minimum distance between any two distinct codewords in the code. Thus, \( d(C) = \min \{ d(x, y) | x \neq y, x, y \in C \} = \min \{ w(x - y) | x \neq y, x, y \in C \} \).

**Remark 2.6.** The Hamming distance satisfies the following properties for any codewords \( x, y, z \in \mathbb{F}_q^n \):

1. (Identity of Indiscernibles) \( d(x, y) = 0 \) if and only if \( x = y \)
2. (Symmetry) \( d(x, y) = d(y, x) \)
3. (Triangle Inequality) \( d(x, y) \leq d(x, z) + d(z, y) \).

Thus, the Hamming distance is a metric on the set \( \mathbb{F}_q^n \).

**Definition 2.7.** An \((n, M, d)\) code is a code that has minimum distance \( d \) and consists of \( M \) codewords, all of length \( n \).

### 2.2. Correcting and Detecting Errors.

The number of errors in a received codeword is the distance between the received codeword and the transmitted word. Suppose a codeword \( x = x_0x_1 \cdots x_{n-1} \) is sent and the codeword \( y = y_0y_1 \cdots y_{n-1} \) is received. Then the error vector is \( e := y - x = e_0e_1 \cdots e_{n-1} \) where \( e_i = y_i - x_i \).

Many codes use a nearest neighbor decoding scheme which decodes a codeword by choosing a word that minimizes the distance between the received word and the possible transmitted word. For a \( q \)-ary code, this scheme will maximize the likelihood of correcting errors provided the following assumptions:

1. Errors in each symbol are independent and identically distributed.
2. If a symbol is received in error, then each of the \( q - 1 \) possible errors is equally likely.

The nearest neighbor decoding scheme will be used for the remainder of the paper.

**Example 2.8.** Suppose we want to transmit two digits, 1 and 0. However, instead of sending a single bit, we send 11111 to represent 1 and 00000 to represent 0. The advantage of this repetition code over single bits is that there is a higher chance for one bit of code to be corrupted by noise, causing unintended message to be received. When using this repetition code above, we may still decode the 5-tuple correctly using the nearest neighbor decoding scheme even if 0 to 2 errors occur.

We can also calculate the probability of decoding a word correctly in this coding scheme if we assume (1), and (2):

Let \( p \) represent the probability of a symbol being received in error. Then the probability of receiving a completely correct 5-tuple is \( (1 - p)^5 \). The probability for one error to occur is \( 5p(1 - p)^4 \). The probability for two errors to occur is
Thereof, the probability of decoding a word correctly is \( (1 - p)^5 + 5p(1 - p)^4 + 10p^2(1 - p)^3 \).

We will now state and prove a result that gives an upper bound on the number of errors that may be detected and corrected in terms of the code's minimum Hamming distance.

**Theorem 2.9.**

1. A code \( C \) can detect up to \( s \) errors in any codeword if \( d(C) = s + 1 \).
2. A code \( C \) can correct up to \( t \) errors in any codeword if \( d(C) = 2t + 1 \) or \( d(C) = 2t + 2 \).

**Proof.**

1. Suppose \( d(C) = s + 1 \). Suppose a codeword \( c \) is transmitted and that \( s \) or fewer errors occur during the transmission. Then, the received word can only be \( c \), since all other codewords are at least different from \( c \) in \( s + 1 \) places. Therefore, the errors are detected. Now we show that more than \( s \) errors might not be detected. Choose \( x, y \in C \) such that \( d(x, y) = s + 1 \). If \( x \) is transmitted as \( y \), then \( s + 1 \) errors have occurred, these errors will not be detected. Thus, the code \( C \) can detect up to \( s \) errors.

2. Suppose \( d(C) = 2t + 1 \) or \( d(C) = 2t + 2 \). Suppose a codeword \( x \) is transmitted and that the received word, \( y \), contains \( t \) or fewer errors. Then \( d(x, y) \leq t \). Let \( x' \) be any codeword other then \( x \). Then \( d(x', y) \geq t + 1 \). Otherwise \( d(x', y) \leq t \), which implies that \( d(x, x') \leq d(x, y) + d(x', y) \leq 2t \) (by the triangle inequality). This is a contradiction to the assumption that \( d(C) = 2t + 1 \) or \( d(C) = 2t + 2 \). Therefore, \( d(x, y) < d(x', y) \). Hence, \( x \) is the nearest codeword to \( y \), so \( y \) is decoded correctly.

Now we show that more than \( t \) errors might not be corrected. Suppose \( d(C) = 2t + 1 \). Let \( x, y \) be two codewords such that \( d(x, y) = 2t + 1 \). For simplicity, assume that the first \( 2t + 1 \) digits of \( x \) and \( y \) differ. Suppose \( x \) is transmitted and the first \( t + 1 \) digits of \( x \) are corrupted into the first \( t + 1 \) digits of \( y \) during transmission. Therefore, the transmitted codeword is \( x_1x_2\cdots x_{t+1}\cdots x_n \), and the received codeword is \( y_1y_2\cdots y_{t+1}x_{t+2}\cdots x_n \). The nearest neighbor decoding scheme will decode the received word as \( y \) but not \( x \), since \( y \) is of distance \( t \) to the received word and \( x \) is of distance \( t + 1 \) to the received word. So, when \( d(C) = 2t + 1 \), the code can correct up to \( t \) errors. Suppose \( d(C) = 2t + 2 \). Let \( a, b \) be two codewords such that \( d(a, b) = 2t + 2 \). For simplicity, assume that the first \( 2t + 2 \) digits of \( a \) and \( b \) differ. Suppose \( a \) is transmitted and the first \( t + 1 \) digits of \( a \) are corrupted into the first \( t + 1 \) digits of \( b \) during transmission. Then \( a \) and \( b \) are both of distance \( t + 1 \) to the received word. Therefore, the errors might not be corrected. So when \( d(C) = 2t + 2 \), the code can correct up to \( t \) errors.

\( \square \)

Notice that in Example 2.8, the repetition code has a minimum distance of 5. By Theorem 2.9, it can detect up to 4 errors and correct up to 2 errors.

**Example 2.10.** The ISBN code is 1-error detecting but not 2-error detecting.

**Proof.** We will first show that ISBN code is 1-error detecting. We do this by showing that changing one position in a valid ISBN does not give another valid
ISBN. Suppose that \( u_1 u_2 u_3 u_4 u_5 u_6 u_7 u_8 u_9 u_{10} \), is a valid ISBN, and suppose the digit \( u_{11-k} \) where \( k \in \{1,2, \cdots 10\} \) is changed to \( x \neq u_{11-k} \). To check the ISBN we compute:

\[
y = u_1 + 2u_2 + \cdots (11-k) x \cdots 9 u_9 + 10 u_{10}.
\]

We know that

\[
u_1 + 2u_2 + \cdots (11-k) u_{11-k} \cdots 9 u_9 + 10 u_{10} \equiv 0 \mod 11.
\]

Subtracting this from \( y \) we see that

\[
y \equiv (11-k)(x - u_{11-k}) \mod 11.
\]

Since \( x \) and \( u_{11-k} \) are distinct and \( k < 11 \), \( y \neq 0 \mod 11 \). Therefore the error is detected.

We can find a counterexample to show that the ISBN code is not 2-error detecting. In Example 2.2, we showed that 0-13165-332-6 is a valid ISBN. We can change two positions in the code to let it become 0-23165-333-6 and show it is a valid ISBN by calculating the weighted check sum:

\[
2(2) + 3(3) + 4(1) + 5(6) + 6(5) + 7(3) + 8(3) + 9(3) + 10(6) = 209 \equiv 0 \mod 11.
\]

Therefore, the ISBN code is not 2-error detecting. Notice by Theorem 2.9, the minimum distance of ISBN code is 2.

\[\square\]

2.3. Linear Codes.

Definition 2.11. A linear code of length \( n \) over \( \mathbb{F}_q \) is a subspace of the vector space \( \mathbb{F}_q^n \). When we are viewing \( \mathbb{F}_q^n \) as a vector space, we will write \( V(n,q) \). If \( C \) is a \( k \)-dimensional subspace of \( V(n,q) \), we say that \( C \) is an \([n,k,d]\) or \([n,k]\) linear code. So, \( C \subset V(n,q) \) if and only if:

1. \( 0 \in C \)
2. if \( x, y \in C \), then \( x + y \in C \)
3. if \( x \in C \) and \( \lambda \in \mathbb{F}_q \), then \( \lambda x \in C \).

Theorem 2.12. The minimum distance, \( d(C) \), of a linear code \( C \) is equal to \( w^*(C) \), the weight of the lowest-weight nonzero codeword.

Proof. There exist codewords \( x, y \) in \( C \) such that \( d(C) = d(x, y) \). By the definition of Hamming distance, we can rewrite this as \( d(C) = w(x - y) \). By the definition of linear code, \( C \) is a subspace of \( V(n,q) \), so the linear combination of any set of codewords in \( C \) is a codeword in \( C \). Therefore, \( x - y \) is a codeword in \( C \). By the definition of \( w^*(C) \), we have \( d(C) = w(x - y) \geq w^*(C) \).

On the other hand, there exists some codeword \( c \in C \) such that \( w^*(C) = w(c) = d(c, 0) \geq d(C) \), since \( 0 \in C \). Since we have shown that both \( d(C) \geq w^*(C) \) and \( d(C) \leq w^*(C) \), we conclude that \( d(C) = w^*(C) \).
Since linear codes are vector spaces, an \([n, k]\) linear code can be specified by giving a basis of \(k\) codewords. Therefore, the code can be generated from linear combinations of the basis vectors.

**Definition 2.13.** A \(k \times n\) matrix \(G\) whose rows form a basis for an \([n, k]\) linear code is called a *generator matrix* of the code. Notice the rows of \(G\) have to be linearly independent.

**Example 2.14.** Let \(G\) be the generator matrix:

\[
G = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}.
\]

Then \(G\) generates the linear code \(C\) where \(C = \{00000, 10001, 01010, 00111, 11011, 10110, 01101, 11100\}\).

Notice that \(C\) is a \([5, 3]\) linear code over \(F_2\). Each codeword has length 5, and the codewords form a subspace of \(F_2^5\). From Theorem 2.12 we also know that \(d(C) = 2\). Therefore, \(C\) is a \((5, 8, 2)\) code.

We now define what it means for two linear codes to be equivalent.

**Definition 2.15.** Two \(q\)-ary codes are called *equivalent* if one can be obtained from the other by a combination of operations of the following types:

1. Permutation of the positions of the codewords.
2. Multiplication of the symbols appearing in a fixed position by a nonzero scalar (i.e. elements of \(F_q\)).

From linear algebra, we see that two \(k \times n\) matrices generate equivalent \([n, k]\) codes if one matrix can be obtained by another by:

1. Permutation of the rows
2. Multiplication of a row by a nonzero scalar
3. Addition of a scalar multiple of one row to another
4. Permutation of the columns
5. Multiplication of any column by a nonzero scalar.

**Example 2.16.** Consider the matrix \(G\) in Example 2.14. If we multiply any rows or columns by a nonzero scalar in \(F_2\), in this case 1, the rows and columns remain the same, so the same code \(C\) is generated. We can also obtain the same code by permuting the rows or columns, or adding one row to another. The matrix \(G'\) below is obtained by adding the third row of \(G\) to the first, and the matrix \(G''\) is obtained by swapping the first and the last column of \(G\). They all generate the same code \(C = \{00000, 10001, 01010, 00111, 11011, 10110, 01101, 11100\}\):

\[
G' = \begin{bmatrix}
1 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1
\end{bmatrix}.
\]
\[ G'' = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}. \]

**Definition 2.17.** Given a linear \([n, k]\) code \(C\), the dual code of \(C\), denoted \(C^\perp\), is the set of vectors of \(V(n, q)\) which are orthogonal to every codeword in \(C\):

\[ C^\perp := \{ v \in V(n, q) \mid v \cdot u = 0, \forall u \in C \}. \]

**Theorem 2.18.** Let \(C\) be a linear code, and let \(H\) be a generator matrix for \(C^\perp\), the dual code of \(C\). Then a vector \(c\) is a codeword in \(C\) if and only if \(cH^T = 0\), or equivalently, if and only if \(HC^T = 0\).

**Proof.** Let \(c \in C\). Then \(c \cdot h = 0\) for all \(h \in C^\perp\). It follows that \(cH^T = 0\), since the rows of \(H\) form a basis for \(C^\perp\).

Alternatively, let \(c\) be a vector such that \(cH^T = 0\). Then \(c \cdot h\) for all \(h\) in the dual code \(C^\perp\). So \(c \in (C^\perp)^\perp\). From linear algebra we know \((C^\perp)^\perp = C\), hence \(c \in C\). \(\square\)

**Definition 2.19.** Let \(C\) be an \([n, k]\) linear code. A parity check matrix for \(C\) is an \((n - k) \times n\) matrix \(H\) such that \(c \in C\) if and only if \(cH^T = 0\). Equivalently, a parity check matrix for \(C\) is a generator matrix for \(C^\perp\).

**Theorem 2.20.** Let \(C\) have a parity check matrix \(H\). The minimum distance of \(C\) is equal to the minimum nonzero number of columns in \(H\) for which a nontrivial linear combination of the columns sums to zero.

**Proof.** Since \(H\) is a parity check matrix for \(C\), \(c \in C\) if and only if \(0 = cH^T\). Let the column vectors of \(H\) be \(d_0, d_1, \ldots, d_{k-1}\). The matrix equation \(0 = cH^T\) can be expressed as follows:

\[ 0 = cH^T = (c_0, c_1, \ldots, c_{k-1})[d_0, d_1, \ldots, d_{k-1}] = c_0d_0 + c_1d_1 + \cdots + c_{k-1}d_{k-1} \]

This shows that \(w(c) > 0\) if and only if there is a nontrivial linear combination of \(d\) columns of \(H\) which equals zero. It now follows from Theorem 2.12 that the minimum distance of \(C\) is equal to the minimum nonzero number of columns in \(H\) for which a nontrivial linear combination of the columns sums to zero. \(\square\)

2.4. Hamming code.

**Definition 2.21.** A binary Hamming code \(H_r\) of length \(n = 2^r - 1, r \geq 2\), is a linear code whose parity check matrix \(H\) has columns consists of all nonzero binary vectors of length \(r\), each used once.

**Example 2.22.** Suppose \(H_3\) is a binary Hamming code. Then by Definition 2.21, \(n = 7\). The parity check matrix of \(H_3\) is:

\[ H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}. \]
Proposition 2.23. The Hamming code, \( H_r \), mentioned above gives an
\([n = 2^r - 1, k = 2^r - 1 - r, d = 3]\) linear code.

Proof. We already know that \( H_r \) has length of \( n = 2^r - 1 \). Since the parity check
matrix, \( H \), of \( H_r \) is a \((n-k)\times n\) matrix, so its generator matrix is a \( k \times n \) matrix.
We know the columns of \( H \) are of length \( r \), so \( H \) has \( r \) rows. This means that
\( n - k = r \). Therefore, \( k = n - r = 2^r - 1 - r \).

We know \( H \) cannot have a zero column, and each column is distinct. Since \( H \)
has binary vectors as its columns, we need at least 3 columns to create a nontrivial
linear combination that sums to zero. Furthermore, we know \( H \) has at least 2 rows
since \( r = n - k \geq 2 \). Let \( x \) be the column of \( H \) that has 1 in the first row but 0
elsewhere. Let \( y \) be the column of \( H \) that has 1 in the second row but 0 elsewhere.
Then let \( z \) be the column of \( H \) that has 1 in both the first and second row but 0
elsewhere. Notice, \( x + y - z = 0 \). Therefore, by Theorem 2.20, \( d = 3 \). \( \square \)

3. Important Results From Algebra

In §3.1, we will review basic results from abstract algebra. In §3.2 we will
introduce Cyclic codes and their correspondence to polynomials. To finish, we
prove a result that relates Hamming codes and Cyclic codes.

3.1. Abstract Algebra Review.

Definition 3.1. A ring is a set \( S \) together with two binary operations + and \( * \)
satisfying the following conditions:

1. \( S \) is an abelian group under addition.
2. Associativity of multiplication: For \( a, b, c \in S \), \( a * (b * c) = (a * b) * c \).
3. Existence of multiplicative identity: There exists \( e \in S \) such that for any
   \( x \in S \), \( e * x = x \).
4. Left and right distributivity: For all \( a, b, c \in S \), \( a * (b + c) = (a * b) + (a * c) \)
   and \( (b + c) * a = (b * a) + (c * a) \).

Examples of rings include \( \mathbb{Z}, \mathbb{F}_q \) where \( q \) is prime, and \( \mathbb{F}_2[x]/(x^7 - 1) \).

Definition 3.2. A nonempty subset \( A \) of a ring \( R \) is an two-sided ideal of \( R \) if
\( a, b \in A \) then \( a + b \in A \). And if \( a \in A \) and \( r \in R \) then \( ra, ar \in A \). When \( R \) is
commutative, \( ar = ra \), hence we only need to check that \( ra \in A \).

Example 3.3. For any ring \( R \), \( \{0\} \) and \( R \) are ideals of \( R \). For the ring \( \mathbb{Z} \), the set
\( n\mathbb{Z} := \{0, \pm n, \pm 2n \cdots \} \) is an ideal if \( n \in \mathbb{Z}^+ \).

Definition 3.4. Let \( R \) be a commutative ring with unity and let \( g \in R \). The set
\( \langle g \rangle = \{rg \mid r \in R \} \) is an ideal of \( R \) called the principal ideal generated by \( g \). The
element \( g \) is called the generator of the principal ideal. So \( A \) is a principal ideal if
there exists \( g \in A \) such that every element \( a \in A \) can be written as \( rg \) for some
\( r \in R \).

Quotient rings of a polynomial ring of the form \( \mathbb{F}[x]/m(x) \) where \( m(x) \) is a
polynomial are of particular importance to us. An important result in abstract
algebra says that \( \mathbb{F}[x] \) is a principal ideal domain (PID). Thus, every ideal \( I \subset
\mathbb{F}[x]/m(x) \) can be generated by a single polynomial. When \( I \) is nonzero, we may
choose the polynomial to have leading coefficient 1. This type of polynomial has a special name: Monic Polynomials.

Definition 3.5. A monic polynomial is a polynomial whose highest-degree coefficient equals 1.

Example 3.6. Polynomials of the form $x^n - 1$ are monic polynomials.

Theorem 3.7. Let $A$ be an ideal in $\mathbb{F}_q[x]/(x^n - 1)$. The following statements are true:

1. There exists a unique monic polynomial $g(x)$ of minimal degree such that $\langle g(x) \rangle \in A$.
2. The ideal $A$ is principal with generator $g(x)$.
3. The polynomial $g(x)$ divides $x^n - 1$ in $\mathbb{F}_q[x]$.

In order to prove this theorem, we need the following lemma:

Lemma 3.8. Suppose that $A$ is an ideal in $\mathbb{F}_q[x]/(x^n - 1)$, and choose a monic polynomial $g(x) \in \mathbb{F}_q[x]$ of minimal degree such that $\langle g(x) \rangle \in A$. If $f(x) \in \mathbb{F}_q[x]$ is any polynomial such that $\langle f(x) \rangle \in A$, then $g(x)$ divides $f(x)$.

Proof. Since $f(x), g(x) \in \mathbb{F}_q[x]$, we can write $f(x) = q(x)g(x) + r(x)$ where $r(x) \neq 0$ and $\deg r(x) < \deg g(x)$. Then $\langle f(x) \rangle = \langle q(x) \rangle \langle g(x) \rangle + \langle r(x) \rangle$. Since $\langle f(x) \rangle \in A$ and $\langle q(x) \rangle \langle g(x) \rangle \in A$, we have $\langle r(x) \rangle \in A$. So, $\langle cr(x) \rangle \in A$ for any scalar $c$. If $r(x) \neq 0$, we can scale it to make it monic, contradicting the minimality of $g(x)$. Hence, $r(x) = 0$ and $g(x)$ divides $f(x)$.

Items 1, 2, and 3 of Theorem 3.7 all follow from the above lemma:

Proof. (1) Suppose that $h(x)$ is another monic polynomial of minimal degree such that $\langle h(x) \rangle \in A$. Then $g(x)h(x)$ by Lemma 3.8. Since $g(x)$ and $h(x)$ have the same minimal degree, so $h(x) = cg(x)$ where $c$ is a scalar. Furthermore, since $h(x)$ and $g(x)$ are monic, $h(x) = g(x)$. Hence, there is a unique monic polynomial $g(x)$ of minimal degree such that $\langle g(x) \rangle \in A$.

(2) Suppose $\langle f(x) \rangle \in A$, then $g(x) \mid f(x)$ by Lemma 3.8. Hence $\langle g(x) \rangle \mid \langle f(x) \rangle$, which implies that $\langle f(x) \rangle = \langle g(x) \rangle \langle g(x) \rangle$ for some $\langle g(x) \rangle \in \mathbb{F}_q[x]/(x^n - 1)$. Thus, $A \subseteq \langle \langle g(x) \rangle \rangle$. Now suppose that $\langle h(x) \rangle \in \langle \langle g(x) \rangle \rangle$. Then, $\langle h(x) \rangle = \langle g(x) \rangle \langle r(x) \rangle$ for some $\langle r(x) \rangle \in \mathbb{F}_q[x]/(x^n - 1)$. Since $\langle g(x) \rangle \in A$, $\langle r(x) \rangle \in \mathbb{F}_q[x]/(x^n - 1)$, the definition of an ideal implies that $\langle h(x) \rangle \in A$. It follows that $\langle g(x) \rangle \subseteq A$. Since we have both inclusions, we conclude that $A = \langle g(x) \rangle$, hence $A$ is a principal ideal with generator $g(x)$.

(3) Note that $\langle x^n - 1 \rangle = 0 \in A$, so by the Lemma 3.8, $g(x)$ must divide $x^n - 1$.

Definition 3.9. A polynomial $g(x)$ is irreducible over the field $\mathbb{F}$ if $g(x)$ cannot be factored into a product of two non-constant polynomials over $\mathbb{F}$.

For references on the two results below, see [6] and or [7].

Theorem 3.10. Let $p(x)$ be an irreducible polynomial of degree $r$ in $\mathbb{F}_q[x]$. Then the ring $\mathbb{F}_q[x]/p(x)$ of polynomials modulo $p(x)$ is actually a field of order $q^r$.

Theorem 3.11. Primitive Element Theorem: The multiplicative group of any finite field is cyclic. This means that for a finite field $\mathbb{F}_q$, there is an $a \in \mathbb{F}_q$ such that $\mathbb{F} = \{0, 1, a, a^2, \ldots, a^{q-2}\}$. 

3.2. Cyclic Codes.

**Definition 3.12.** An \([n, k, d]\) linear code \(C\) is cyclic if whenever \((c_0, c_1, \ldots, c_{n-1})\) is a codeword in \(C\), then \((c_{n-1}, c_0, \ldots, c_{n-2})\) is also a codeword in \(C\).

**Example 3.13.** The binary code \(C = \{000, 101, 011, 110\}\) is a cyclic code.

There is a correspondence between codes in \(C \subseteq \mathbb{F}_q^n\) and sets of polynomials \(C' \subseteq \mathbb{F}_q[x]/(x^n - 1)\). Given a code, \(C \subseteq \mathbb{F}_q^n\), we define

\[
C' := \{[c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}]\} \in \mathbb{F}_q[x]/(x^n - 1)
\]

such that \((c_0, \ldots, c_{n-1}) \in C\). Conversely, given a subset \(C' \subseteq \mathbb{F}_q[x]/(x^n - 1)\), we can define a code \(C := \{(c_0, \ldots, c_{n-1})\} \in \mathbb{F}_q^n\) such that \([c_0 + \ldots + c_{n-1} x^{n-1}] \in C'\). This correspondence is well-defined and gives a bijection between sets. When \(C\) is a cyclic code, \(C'\) has special structure.

We can achieve a right cyclic shift of a codeword by multiplying its associated code polynomial, \(c(x) = c_0 + c_1 x + \ldots + c_{n-1} x^{n-1}\), by \(x\):

\[
c'(x) = c_0 x + c_1 x^2 + \ldots + c_{n-1} x^n \equiv c_0 x + c_1 x^2 + \ldots + c_{n-1} \mod x^n - 1.
\]

The codeword associated with \(c'(x)\) is \((c_{n-1}, c_0, \ldots, c_{n-2})\), which is the right shift of the codeword associated with \(c(x)\): \((c_0, c_1, \ldots, c_{n-1})\).

We will now use code polynomials to characterize cyclic codes:

**Theorem 3.14.** The linear code of length \(n\) over \(\mathbb{F}_q\) is cyclic if and only if \(C'\) satisfies the following two conditions:

1. If \(a(x), b(x) \in C\) then \(a(x) + b(x) \in C'\)
2. If \(a(x) \in C'\) and \(r(x) \in \mathbb{F}_q[x]/(x^n - 1)\), then \(r(x) a(x) \in C'\).

**Proof.** Suppose \(C\) is a cyclic code of length \(n\) over \(\mathbb{F}_q\). Then \(C\) is linear, so condition (1) holds. Now suppose that \(a(x) \in C'\) and 

\[
r(x) = r_0 + r_1 x + \cdots + r_{n-1} x^{n-1} \in \mathbb{F}_q[x]/(x^n - 1).
\]

As discussed above, multiplication of a code polynomial by \(x\) corresponds to a cyclic shift of the corresponding codeword. Hence, \(x a(x) \in C'\), \(x (a(x)) = x^2 a(x) \in C'\), and so on. It follows that \(r(x) a(x) = r_0 a(x) + r_1 x a(x) + \cdots + r_{n-1} x^{n-1} a(x)\) is also in \(C'\) since each summand is in \(C'\). Therefore, condition (2) also holds.

On the other hand, suppose that conditions (1) and (2) hold. If we take \(r(x)\) to be a scalar, the conditions imply that \(C\) is a linear code. Then, if we take \(r(x) = x\), condition (2) implies that \(C\) is a cyclic code.

**Corollary 3.15.** Cyclic codes of length \(n\) over \(\mathbb{F}_q\) are precisely the ideals in the ring \(\mathbb{F}_q[x]/(x^n - 1)\).

**Proof.** Suppose \(C\) is a cyclic code of length \(n\) over \(\mathbb{F}_q\) and let \(C'\) be its set of code polynomials in \(\mathbb{F}_q[x]/(x^n - 1)\). By Theorem 3.14, \(C'\) is an ideal in \(\mathbb{F}_q[x]/(x^n - 1)\).

On the other hand, suppose that \(A'\) is an ideal in \(\mathbb{F}_q[x]/(x^n - 1)\) and let \(A\) be the corresponding code in \(\mathbb{F}_q^n\). Then the elements of \(A'\) are polynomials of degree less than or equal to \(n - 1\), and the set of polynomials is closed under linear combinations. This shows that \(A\) is a subspace of \(\mathbb{F}_q^n\). Furthermore, by Definition 3.2, if \(a(x) \in A'\), then \(r(x) a(x) \in A'\) for any polynomial \(r(x) \in \mathbb{F}_q[x]/(x^n - 1)\). This implies that \(A\) is a cyclic code.
We can actually translate Theorem 3.7 into the language of coding theory:

**Theorem 3.16.** Let $C$ be a $q$-ary $[n,k]$ cyclic code.

1. In $C$, there is a unique monic code polynomial $g(x)$ with minimal degree $r < n$ called the generator polynomial of $C$.
2. Every code polynomial $c(x) \in C$ can be expressed uniquely as
   
   $$c(x) = m(x)g(x),$$
   
   where $g(x)$ is the generator polynomial and $m(x)$ is a polynomial of degree less than $(n-r)$ in $\mathbb{F}_q[x]$.
3. The generator polynomial $g(x)$ divides $(x^n - 1)$ in $\mathbb{F}_q[x]$.

**Theorem 3.17.** Suppose $C$ is a cyclic code with generator polynomial $g(x) = g_0 + g_1 x + \cdots + g_r x^r$ of degree $r$. Then the dimension of $C$ is $n-r$, and a generator matrix for $C$ is the following $(n-r) \times n$ matrix:

$$G = \begin{bmatrix}
g_0 & g_1 & \cdots & g_r & 0 & 0 & \cdots & 0 \\
0 & g_0 & g_1 & \cdots & g_r & 0 & \cdots & 0 \\
0 & 0 & g_0 & g_1 & \cdots & g_r & \cdots & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & g_0 & g_1 & \cdots & g_r
\end{bmatrix}.$$

**Proof.** First, note that $g_0$ is nonzero: Otherwise, $(0, g_1, \cdots, g_{r-1}) \in C$ which implies that $(g_1, \cdots, g_{r-1}, 0) \in C$ which implies that $g_1 + g_2 x + \cdots + g_{r-1} x^{r-1} \in C$, which contradicts the minimality of the degree $r$ of the generating polynomial. Now, we see that the $n-r$ rows of the matrix $G$ are linearly independent because of the echelon of nonzero $g_0$s with $0$s below. These $n-r$ rows represent the code polynomials $g(x), x g(x), x^2 g(x), \cdots, x^{n-r-1} g(x)$. In order to show that $G$ is a generator matrix for $C$ we must show that every code polynomial in $C$ can be expressed as a linear combination of $g(x), x g(x), x^2 g(x), \cdots, x^{n-r-1} g(x)$. Part 2 of Theorem 3.16 shows that if $c(x)$ is a code polynomial in $C$, then $c(x) = m(x) g(x)$ for some polynomial $m(x)$ of degree less than $n-r$ in $\mathbb{F}_q[x]$. Hence,

$$c(x) = m(x) g(x) = (m_0 + m_1 x + \cdots + m_{n-r-1} x^{n-r-1}) g(x) = m_0 g(x) + m_1 g(x) + \cdots + m_{n-r-1} x^{n-r-1} g(x)$$

which shows that any code polynomial $c(x)$ in $C$ can be written as a linear combination of the code polynomials represented by the $n-r$ independent rows of $G$. We conclude that $G$ is a generator matrix for $C$ and the dimension of $C$ is $n-r$. \qed

We conclude this paper with a fundamental result relating Cyclic and Hamming codes:

**Theorem 3.18.** The binary Hamming code $H_r$ is equivalent to a Cyclic code.

**Proof.** Let $p(x)$ be an irreducible polynomial of degree $r$ in $\mathbb{F}_2[x]$. Then, the ring $\mathbb{F}_2[x]/p(x)$ of polynomials modulo $p(x)$ is a field of order $2^r$ by Theorem 3.10. Since every finite field has a primitive element (by Theorem 3.11), there exists an element $\alpha \in \mathbb{F}_2[x]/p(x)$ such that $\mathbb{F}_2[x]/p(x) = \{0, 1, \alpha, \alpha^2, \cdots, \alpha^{2^r-2}\}$. Consider the matrix $H = \{[1, \alpha, \cdots, \alpha^{2^r-2}]\}$. Notice each entry of $H$ is an element of the field $\mathbb{F}_2[x]$, the ring $\mathbb{F}_2[x]/p(x)$. Therefore, $H$ is a generator matrix for the binary Hamming code $H_r$. The binary Hamming code $H_r$ is equivalent to a Cyclic code.
which is of order $2^r$. Thus, we can express each entry $\alpha^i$ with $0 \leq i \leq 2^r - 1$ as a binary column vector $(a^i_0, a^i_1, \cdots, a^i_{r-1})^T$ where

$$\alpha^i = a^i_0 + a^i_1 x + \cdots + a^i_{r-1} x^{r-1}.$$ 

This allows us to think of $H$ as an $r \times (2^r - 1)$ binary matrix.

Let $C$ be the linear code having $H$ as its parity check matrix. Since the columns of $H$ are exactly the $2^r - 1$ nonzero binary vectors of length $r$, $C$ is a length $n = 2^r - 1$ Hamming code.

We now show that $C$ is cyclic. Since $H$ is the parity check matrix for $C$, we have $c \in C$ if and only if $c^H^T = 0$ by Theorem 2.18. Therefore:

$$C = \{(c_0, c_1, \cdots, c_{n-1}) \in V(n, 2) \mid c_0 + c_1 \alpha + \cdots + c_{n-1} \alpha^{n-1} = 0\}$$

$$C' = \{c(x) \in F_2[x]/(x^n - 1) \mid c(\alpha) = 0 \in F_2[x]/p(x)\}.$$

If $c(x) \in C'$ and $r(x) \in F_2[x]/p(x)$, then $r(x)c(x) \in C'$ because $r(\alpha)c(\alpha) = r(\alpha) \cdot 0 = 0$. It follows from Theorem 3.14 that $C$ is cyclic. □

Acknowledgments

It is a pleasure to thank my mentor, Katie Gallagher, for helping me to understand basic abstract algebra and giving me suggestions for this paper.

References

[6] Lindsay N. Childs. “A Concrete Introduction to Higher Algebra”