FACULTAD DE **MATEMÁTICAS**

UNIVERSIDAD DE **MURCIA**

Grado en Matemáticas

Trabajo Fin de Grado

TÉCNICAS ALGEBRAICAS EN LA DECODIFICACIÓN DE CÓDIGOS CORRECTORES DE ERRORES

Realizado por Francisco Javier Marín Ruiz

Dirigido por Juan Jacobo Simón Pinero

Junio 2018

Declaración de originalidad

Francisco Javier Marín Ruiz, autor del Trabajo Fin de Grado *Técnicas Algebraicas en la Decodificación de Códigos Correctores de Errores*, bajo la tutela del profesor Juan Jacobo Simón Pinero, declara que el trabajo que presenta es original, en el sentido de que ha puesto el mayor empeño en citar debidamente todas las fuentes utilizadas.

En Murcia, a veintiséis de junio de dos mil dieciocho.

Fdo.: Francisco Javier Marín Ruiz.

Índice general

Introducción

En términos generales, la teoría de códigos es el estudio de métodos para que la transmisión de información de un lugar a otro sea lo más eficiente, segura y precisa posible. Esta transmisión siempre se lleva a cabo a través de lo que llamaremos un *canal de comunicación* o simplemente *canal* (como puede ser el aire o una línea telefónica), y la información está formada "inicialmente" por mensajes, los cuales están formados por palabras y éstas, a su vez, por símbolos. Al conjunto de símbolos $\mathcal{A}$ con el que formamos las palabras lo llamaremos *alfabeto* y al conjunto de palabras con el que formamos mensajes lo denotaremos por $Pal(\mathcal{A})$. Un ejemplo de alfabeto es el abecedario latino y un ejemplo de un elemento de $Pal(\mathcal{A})$ es cualquier palabra del diccionario o, aún más simple, cualquier símbolo del alfabeto. Las comillas en el término 'inicialmente' hacen referencia a la codificación que se le aplica a cualquier mensaje antes de ser enviado. *Codificar* un mensaje es reescribirlo, ya sea mediante los símbolos del mismo alfabeto (como veremos más adelante), ya sea mediante símbolos de otro alfabeto, digamos $\mathcal{F}$. Siendo más concretos, codificar un mensaje (finito) $M \subseteq Pal(\mathcal{A})$ es aplicar a M una función inyectiva $\mathcal{C} : Pal(\mathcal{A}) \to Pal(\mathcal{F})$. Con esto ya podemos explicar mejor el motivo de las comillas: la información está formada en todo momento por mensajes, pero normalmente es sólo al inicio cuando estos mensajes están escritos con el alfabeto $\mathcal{A}$.

Dado el subconjunto finito $S \subseteq Pal(\mathcal{A})$ con el cual se establecerá una comunicación, llamaremos *código* a $\mathcal{C}(S)$. Un ejemplo de código es el conocido código ASCII. Aquí:

- El alfabeto $\mathcal{A}$ está formado por el abecedario latino y una serie de caracteres como el espacio, los signos de interrogación, etc.
- El alfabeto $\mathcal{F}$ es $\mathbb{Z}_2$.
- El subconjunto $S \subseteq Pal(\mathcal{A})$ es el propio alfabeto $\mathcal{A}$: cada símbolo a de $\mathcal{A}$ constituye una palabra s de S y a cada uno de estos símbolos se le asigna por $\mathcal{C}$ una palabra formada por ocho símbolos de $\mathcal{F}$.

Así, si por ejemplo

$$h \overset{\mathcal{C}}{\rightsquigarrow} 01010101 \qquad o \overset{\mathcal{C}}{\rightsquigarrow} 00000000 \qquad l \overset{\mathcal{C}}{\rightsquigarrow} 11111111 \qquad a \overset{\mathcal{C}}{\rightsquigarrow} 10101010,$$

el mensaje *hola* se codificaría como 0101010100000000111111110101010.

Llamaremos *palabra del código* o simplemente *palabra* a cada elemento de $\mathcal{C}(S)$. Debido a las ventajas que ofrecen los cuerpos finitos, el alfabeto $\mathcal{F}$ en el que se codificarán las

palabras será siempre un cuerpo finito de q elementos $\mathbb{F}_q$, donde q es una potencia de un número primo p.

Por "regla de tres", uno puede pensar que decodificar una palabra es reescribirla con los símbolos del alfabeto inicial. Sin embargo, entenderemos que *decodificar* una palabra recibida es reemplazarla por aquella palabra del código que creemos que ha sido enviada. Veamos un ejemplo en el que, de paso, mencionamos el principal problema de muchos canales de comunicación: imaginemos que estamos estableciendo una comunicación con otra persona mediante el abecedario latino como alfabeto $\mathcal{A}$ y mediante el conjunto

$$S = \{perro, gato, tortuga, liebre\} \subseteq Pal(\mathcal{A})$$

y que, para enviar cualquiera de estas palabras:

(i) Se codifica la palabra como sigue:

perro	*gato*	*tortuga*	*liebre*
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
00	01	10	11

Es entonces $\mathcal{F} = \mathbb{F}_2$.

(ii) Se envía la palabra codificada.

(iii) Se reescribe al alfabeto $\mathcal{A}$ la palabra recibida siguiendo la misma regla.

Si el canal de comunicación estuviera libre de posibles interferencias (llamadas *ruidos*), todos los mensajes se transmitirían correctamente:

$$liebre \xrightarrow{\text{codificamos}} 11 \xrightarrow{\text{enviamos}} \boxed{\text{canal}} \xrightarrow{\text{recibimos}} 11 \xrightarrow{\text{reescribimos}} liebre$$

Sin embargo, el canal puede no estar exento de ruidos, pudiendo ocurrir lo siguiente:

$$liebre \xrightarrow{\text{codificamos}} 11 \xrightarrow{\text{enviamos}} \boxed{\text{canal}} \xrightarrow{\text{recibimos}} 01 \xrightarrow{\text{reescribimos}} gato$$
$$\updownarrow$$
$$\text{ruido}$$

El canal nos ha dado *gato* por *liebre*. Añadiendo *redundancia*, como veremos más adelante, podemos resolver el anterior problema, pero con conjuntos de palabras mucho mayores es inevitable preguntarse qué códigos son óptimos en el sentido de codificar y decodificar de manera rápida además de:

- Permitir enviar la máxima cantidad de información por unidad de tiempo.
- Asegurar la privacidad de los mensajes.
- Detectar y corregir errores.

Responder a estas tres cuestiones es uno de los objetivos de la teoría de códigos y podemos distinguir tres tipos de códigos:

- Códígos compresores.
- Códigos secretos.
- Códigos correctores de errores.

En este trabajo nos ocuparemos de los códigos correctores de errores; en concreto, veremos el trasfondo algebraico que hay detrás de los algoritmos de codificación y decodificación de la información que involucran los códigos correctores de errores.

En el primer capítulo veremos las definiciones más básicas y las suposiciones que haremos sobre los códigos y el canal de comunicación; las nociones de *error*, *redundancia* y *decodificación* por las que se sustentan los códigos correctores; y el concepto de distancia mínima en el contexto de estos códigos, por la cual quedan determinadas las capacidades de detección y corrección de errores de los mismos.

En el segundo capítulo comenzamos a entrar en materia introduciendo los *códigos lineales*, que son aquellos que pueden identificarse con subespacios vectoriales de un espacio total $\mathbb{F}_q^n$. De esta identificación se derivan, como se podía intuir, los conceptos de ortogonalidad, base y espacio cociente en códigos lineales. También surge una nueva forma de decodificar: la decodificación por *síndrome* (o mediante una *tabla de síndromes*).

De entre todos los códigos lineales, en el tercer capítulo veremos los llamados *códigos cíclicos* y los asociaremos con los ideales del anillo cociente $\mathbb{F}_q[x]/(x^n - 1)$. Estos ideales, a su vez, los asociaremos con un polinomio que, si cumple ciertos requisitos, es único: el *polinomio generador*. A partir de este polinomio veremos una nueva (y más rápida) forma de calcular los síndromes, además de una reducción en el tamaño de la tabla de síndromes.

En el cuarto y último capítulo ahondaremos en los códigos cíclicos para ver los *códigos BCH*, donde intervienen polinomios irreducibles, raíces n-ésimas primitivas de la unidad, etc. Para la decodificación de estos códigos veremos el algoritmo de Sugiyama, una variación desarrollada en 1975 del algoritmo de Peterson-Gorenstein-Zierler (1960) que usa una modificación del algoritmo de Euclides para polinomios.

Para terminar, en el anterior ejemplo podíamos haber codificado de la siguiente manera:

perro	*gato*	*tortuga*	*liebre*
↓	↓	↓	↓
0	1	00	01

Pero lo hicimos con elementos de $\mathbb{F}_2^2$ dado que todos los códigos que manejaremos estarán formados por palabras de una determinada longitud k; es decir, trabajaremos con códigos en bloques.

Introduction

Coding theory is the study of methods to make the transmission of information from one place to another as efficient, secure and accurate as possible. This transmission will be always through what we will call a *communication channel* or just a *channel* (as the air or a telephone line) and the information is composed "initially" by messages, which are composed by words and these words are composed by symbols. We will call *alphabet* the set of symbols $\mathcal{A}$ from which we will make words and we will denote by $Pal(\mathcal{A})$ the set of all words that can be made with the alphabet $\mathcal{A}$. An example of alphabet is the Latin alphabet and an example of an element of $Pal(\mathcal{A})$ is any word of the dictionary or even any symbol of the alphabet. The quotation marks on the word 'initially' refer to the encoding applied to any message before it is sent. *Encoding* a message is to rewrite it either with the symbols of the same alphabet $\mathcal{A}$, either with the symbols of any other alphabet $\mathcal{F}$. To be more clear, encoding a (finite) message $M \subseteq Pal(\mathcal{A})$ is to apply the injective map $\mathcal{C} : Pal(\mathcal{A}) \to Pal(\mathcal{F})$ to M. With this we can now explain better why we put the quotation marks: the information is always composed by messages, but is usually only at the beginning when these are written with the symbols of the alphabet $\mathcal{A}$.

Given a finite subset $S \subseteq Pal(\mathcal{A})$ with which we will establish a communication, we will call $\mathcal{C}(S)$ a *code*. A well known example of code is the ASCII code, where:

- The alphabet $\mathcal{A}$ is composed by the Latin alphabet and other symbols like '?', '!', etc.
- The alphabet $\mathcal{F}$ is $\mathbb{Z}_2$.
- The subset $S \subseteq Pal(\mathcal{A})$ is the alphabet $\mathcal{A}$: every symbol a of $\mathcal{A}$ is a word s of S by itself and each one of these symbols has associated by $\mathcal{C}$ a word made with eight symbols of $\mathcal{F}$.

For example, with

$$h \overset{\mathcal{C}}{\rightsquigarrow} 00000000 \qquad i \overset{\mathcal{C}}{\rightsquigarrow} 11111111$$

we would decode *hi* as 0000000011111111.

We will call *codeword* or just *word* to any element of $\mathcal{C}(S)$. Due to the properties that finite fields have, the alphabet $\mathcal{F}$ that we will use to encode words will be always a finite field $\mathbb{F}_q$ with q elements, where q is a power of a prime number p.

Encoding a word is to rewrite it, but by *decoding* a word we will understand the process of guessing which codeword was sent. Let us see an example where we include the main

problem that many channels have. Let us suppose that we are having a communication using the Latin alphabet as the alphabet $\mathcal{A}$ and the set

$$S = \{dog, donkey, turtle, horse\} \subseteq Pal(\mathcal{A})$$

The communication works as follows:

(i) The word to be sent is encoded:

dog	$donkey$	$turtle$	$horse$
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
00	01	10	11

So it is $\mathcal{F} = \mathbb{F}_2$.

(ii) The encoded word is sent.

(iii) The received word is rewritten with the symbols of the first alphabet $\mathcal{A}$ using (i).

If the channel were *noise free* (i.e., the channel has no interferences), we would always receive the sent codeword:

$$horse \xrightarrow{\text{encode}} 11 \xrightarrow{\text{send}} \boxed{\text{channel}} \xrightarrow{\text{receive}} 11 \xrightarrow{\text{rewrite}} horse$$

However, this can happen if the channel is not noise free:

$$horse \xrightarrow{\text{encode}} 11 \xrightarrow{\text{send}} \boxed{\text{channel}} \xrightarrow{\text{receive}} 01 \xrightarrow{\text{rewrite}} donkey$$

$$\updownarrow$$

$$\text{noise}$$

As we will see, adding what we will call *redundancy* can solve the previous problem. However, with way bigger sets of words the question of which codes are the best codes emerges. By "best codes" we mean those that have a fast encoding and decoding of information and:

- Maximum transmission of information per unit time.
- Maximum privacy.
- Can detect and correct as many errors as possible.

Answering these three questions is one of the purposes of coding theory and there are three types of codes:

- Data compression codes.
- Cryptographic codes.
- Error correcting codes (or error correction codes).

Here we will take care of error correcting codes. To be more specific, we will see the algebra behind the encoding and decoding algorithms that error correcting codes involve.

In the first chapter we will see the basic definitions and assumptions that we will do about codes and the communication channel; the notions of *error*, *redundancy* and *decoding* by which error correcting codes exist; and, in the context of these codes, the meaning of minimum distance, by which their capacity to detect and correct errors are determined.

In the second chapter we will get down to business introducing *linear codes*, that are those codes that can be identified with vector spaces. Therefore, we will talk about orthogonality, basis and quotient spaces in linear codes. We will also see a new way to decode words: the *syndrome decoding*.

Among all linear codes, in the third chapter we will see the *cyclic codes* and we will identify them with the ideals of the residue class ring $\mathbb{F}_q[x]/(x^n-1)$. These ideals will be associated with a polynomial that is unique provided that it satisfies certain conditions: the *generator polynomial*. From this polynomial we will see a new (and faster) way to compute the syndromes, as well as a size reduction of the table of syndromes.

In the fourth and last chapter we will see *BCH codes*, a type of cyclic codes where irreducible polynomials and primitive nth roots of unity come into play. To decode these codes we will see the Sugiyama algorithm, a variation of the Peterson-Gorenstein-Zierler algorithm (1960) developed in 1975 that partially uses the Euclidean Algorithm for polynomials.

To finish this introduction, in the previous example we could have encoded the words of the set S in the following way:

dog	*donkey*	*turtle*	*horse*
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
0	1	00	01

But we did it with elements of $\mathbb{F}_2^2$ because we will only consider codes whose words have the same length k; that is, we will only consider block codes.

Capítulo 1

Detección y corrección de errores

1.1. Preliminares: cuerpos finitos

A lo largo de todo este trabajo denotaremos con q a cualquier potencia de un número primo, digamos p.

Dado un cuerpo K, diremos que las extensiones de cuerpos $K \subset L$ y $K \subset L'$ son K-*isomorfas* si existe un isomorfismo $\sigma : L \to L'$ de forma que $\sigma|_K$ sea la inclusión de K en L'. Se dice en tal caso que σ es un K-*isomorfismo*.

Dados un cuerpo K y un conjunto de polinomio $\mathcal{P} \subseteq K[x]$ con conjunto de raíces S, llamaremos *cuerpo de descomposición* de $\mathcal{P}$ sobre K a una extensión L de K de forma que $L = K(S)$; es decir, de forma que L esté generado sobre K por el conjunto de raíces S.

Las demostraciones de los siguientes tres resultados pueden encontrarse en [1].

Teorema 1.1.1 (Kronecker)**.** *Si K es un cuerpo y $f \in K[x]$ es un polinomio no constante, entonces existe una extensión L de K en la que f tiene una raíz.*

Proposición 1.1.2. *Si K es un cuerpo, entonces cualesquiera dos cuerpos de descomposición de un conjunto de polinomios $\mathcal{P} \subseteq K[x]$ son K-isomorfas.*

Proposición 1.1.3. *Un cuerpo F tiene $q = p^m$ elementos si y sólo si es un cuerpo de descomposición sobre $\mathbb{Z}_p$ del polinomio $x^q - x$.*

Así, si $\alpha_1, \ldots, \alpha_q$ son las raíces de $x^q - x$, por el teorema de Kronecker existe un cuerpo de descomposición $F = \mathbb{Z}_p(\alpha_1, \ldots, \alpha_q)$ de $x^q - x$ sobre $\mathbb{Z}_p$. Es por tanto F un cuerpo finito con q elementos y tal cuerpo es único salvo $\mathbb{Z}_p$-isomorfismos. Lo enunciamos como un corolario.

Corolario 1.1.4. *Para cualesquiera enteros positivos p y m con p primo existe un cuerpo con p^m elementos. Dos cuerpos con p^m elementos son $\mathbb{Z}_p$-isomorfos.*

Con lo cual siempre podremos considerar un cuerpo $\mathbb{F}_q$ con $q = p^m$ elementos.

1.2. Canales de comunicación

Definición 1.2.1. *Sea $\mathbb{F}_q$ un cuerpo de q elementos, al cual llamaremos* alfabeto del código *y a cuyos elementos llamaremos* símbolos del código*:*

(i) Una palabra q-aria *de longitud k sobre $\mathbb{F}_q$ es una sucesión*

$$\mathbf{w} = w_1 \cdots w_k \qquad \text{con } w_i \in \mathbb{F}_q \quad (\forall i \in \{1, \ldots, k\})$$

También podemos interpretar $\mathbf{w}$ como el vector $(w_1, \ldots w_k)$.

(ii) Un código q-ario *de longitud k sobre $\mathbb{F}_q$ es un conjunto no-vacío C de palabras q-arias de longitud k; es decir, es un subconjunto no-vacío $C \subseteq \mathbb{F}_q^k$.*

(iii) A los elementos de C los llamaremos palabras del código *C.*

(iv) Un código de longitud k y tamaño M se llama (k, M)-código.

Denotaremos con $|C|$ al orden de C.

Ejemplos 1.2.2.

(i) Un código sobre $\mathbb{F}_2$ se llama *código binario*; un ejemplo es $C = \{00, 01, 10, 11\}$.

(ii) Un código sobre $\mathbb{F}_3$ se llama *código ternario*; un ejemplo es $C = \{22, 10, 02, 21\}$.

Definición 1.2.3. *Un* canal de comunicación *consiste en un* alfabeto del canal $\mathbb{F}_q = \{a_1, \ldots, a_q\}$ *y en un conjunto de* probabilidades de envío por el canal

$$\mathcal{P}(\text{ haber recibido } a_j \mid \text{habiendo enviado } a_i)$$

satisfaciendo

$$\sum_{j=1}^{q} \mathcal{P}(\text{ haber recibido } a_j \mid \text{habiendo enviado } a_i) = 1 \quad \forall i \in \{1, \ldots, q\}$$

($\mathcal{P}(A|B)$ es la probabilidad condicionada de que suceda A habiendo sucedido B)

Definición 1.2.4. *Decimos que un canal de comunicación* no tiene memoria *(o que es un canal* sin memoria*) si para cada $\mathbf{x}, \mathbf{c} \in \mathbb{F}_q^k$ se tiene*

$$\mathcal{P}(\text{ haber recibido } \mathbf{x} \mid \text{habiendo enviado } \mathbf{c}) = \prod_{i=1}^{k} \mathcal{P}(\text{ haber recibido } x_i \mid \text{habiendo enviado } c_i)$$

Definición 1.2.5. *Un* canal q-ario simétrico *es un canal sin memoria con un alfabeto del canal de tamaño q cumpliendo que*

(i) todo símbolo transmitido tiene la misma probabilidad $p < \frac{1}{2}$ de no recibirse correctamente.

(ii) si un símbolo ha sido recibido por error, entonces las $q-1$ posibles alternativas son equiprobables.

En particular, un canal binario simétrico (BSC) *es un canal sin memoria con alfabeto $\mathbb{F}_2$ y probabilidades del canal*

$$\mathcal{P}(\text{ h.r. } 1 \mid \text{h.e. } 0) = \mathcal{P}(\text{ h.r. } 0 \mid \text{h.e. } 1) = p$$

$$\mathcal{P}(\text{ h.r. } 0 \mid \text{h.e. } 0) = \mathcal{P}(\text{ h.r. } 1 \mid \text{h.e. } 1) = 1 - p$$

donde "h.r." es "haber recibido" y "h.e." es "habiendo enviado". Por tanto, la probabilidad de error en un BSC es p. Esta probabilidad se llama probabilidad cruzada *del BSC.*

Ejemplo 1.2.6. Supongamos que se están enviando palabras del código $\{000, 111\}$ a través de un BSC con probabilidad cruzada $p = 0,05$ y que se ha recibido la palabra 110. Entonces

$$\mathcal{P}(\text{ h.r. } 110 \mid \text{h.e. } 000) = \mathcal{P}(\text{ h.r. } 1 \mid \text{h.e. } 0)^2\ \mathcal{P}(\text{ h.r. } 0 \mid \text{h.e. } 0) = p^2(1-p) = 0,002$$

$$\mathcal{P}(\text{ h.r. } 110 \mid \text{h.e. } 111) = \mathcal{P}(\text{ h.r. } 1 \mid \text{h.e. } 1)^2\ \mathcal{P}(\text{ h.r. } 0 \mid \text{h.e. } 1) = (1-p)^2 p = 0,045$$

y 111 es la palabra más probable a haber sido enviada.

Regla de decodificación

Como dijimos en la introducción, entenderemos por decodificar una palabra recibida al proceso de identificar qué palabra del código fue la enviada. Para llevar esto a cabo, vamos a añadir redundancia, esto es, modificar la codificación (o recodificar) de forma que en las palabras del código haya símbolos "extra" que nos pueda ayudar a determinar qué palabra fue enviada. Así, si S es un subconjunto finito de $Pal(\mathcal{A})$, pasamos del esquema

$$S \xrightarrow{\text{codificamos}} \mathbb{F}_q^k \longrightarrow \boxed{\text{canal}} \xrightarrow{\text{recibimos}} \mathbb{F}_q^k \xrightarrow{\text{reescribimos}} S$$

al esquema

$$S \xrightarrow{\text{codificamos}} \mathbb{F}_q^k \xrightarrow[\text{redundancia}]{\text{añadimos}} \mathbb{F}_q^n \to \boxed{\text{canal}} \to \mathbb{F}_q^n \xrightarrow{\text{decodificamos}} \mathbb{F}_q^k \xrightarrow{\text{reescribimos}} S$$

Un ejemplo de codificación en este sentido es el llamado *dígito de control de paridad*. Consiste en añadir a cada palabra del código un nuevo dígito: la suma módulo 2 de

los dígitos de la palabra en cuestión. Si aplicamos esto al ejemplo de la introducción, obtenemos:

perro	*gato*	*tortuga*	*liebre*
↓	↓	↓	↓
00	01	10	11
↓	↓	↓	↓
000	011	101	110

Nótese que cambiar un dígito arbitrario en cualquier palabra da lugar a una palabra que no está en el código. Por tanto, somos capaces de detectar un error; es decir, el cambio de un dígito en una posición (hablaremos de esto más adelante).

$$\textit{liebre} \longrightarrow 110 \longrightarrow \boxed{\text{canal}} \longrightarrow 010 \longrightarrow ???$$

Sin embargo, en este caso no somos capaces de decidir de qué palabra viene 010 a menos que lo establezcamos arbitrariamente (hablaremos de esto más adelante). Cambiemos el dígito de control de paridad por otro tipo de redundancia: repeticiones, es decir, enviar varias veces la misma palabra. Tenemos:

perro	*gato*	*tortuga*	*liebre*
↓	↓	↓	↓
00	01	10	11
↓	↓	↓	↓
000000	010101	101010	111111

Con este código también podemos detectar un error, pero además somos capaces de decidir de qué palabra viene cualquiera que hayamos recibido con un error eligiendo la palabra del código que difiere en menos posiciones con la palabra recibida.

$$\textit{liebre} \longrightarrow 111111 \longrightarrow \boxed{\text{canal}} \longrightarrow 101111 \overset{\text{decodificamos}}{\longrightarrow} 111111 \longrightarrow \textit{liebre}$$

Éste no es el único criterio de decodificación.

Desde el punto de vista matemático, añadir redundancia es aplicar una función $\mathbb{F}_q^k \to \mathbb{F}_q^n$, así que a partir de ahora no veremos a C como un subconjunto de $\mathbb{F}_q^k$, sino de $\mathbb{F}_q^n$, y nuestro objetivo será decodificar una palabra recibida por la que creamos que ha sido enviada. De hecho, en relación a la Definición 1.2.1 (ii), supondremos de aquí en adelante que $C = \mathbb{F}_q^k$. Con la notación que estamos usando, llamamos *tasa de transmisión* a $\frac{k}{n}$: necesitamos enviar n símbolos para transmitir información de k símbolos.

1.3. Decodificación de máxima verosimilitud

Supongamos que se están enviando palabras de un código $C \subseteq \mathbb{F}_q^n$ a través de un canal de comunicación y que conocemos las probabilidades de envío por el canal

$$\mathcal{P}(\text{ haber recibido } \mathbf{x} \mid \text{habiendo enviado } \mathbf{c}) \quad (\forall \mathbf{c} \in C)$$

Dada una palabra[1] $\mathbf{x} \in \mathbb{F}_q^n$, la *regla de decodificación de máxima verosimilitud (MLD)* establece que la palabra del código más propensa a haber sido enviada es la palabra $\mathbf{c_x} \in C$ tal que

$$\mathcal{P}(\text{ h.r. } \mathbf{x} \mid \text{h.e. } \mathbf{c_x}) = \max_{\mathbf{c} \in C} \mathcal{P}(\text{ h.r. } \mathbf{x} \mid \text{h.e. } \mathbf{c})$$

Hay dos tipos de MLD. Dada una palabra $\mathbf{x}$ recibida, supongamos que encontramos varias palabras del código con misma probabilidad de haber sido enviadas:

- Si eligimos una de ellas arbitrariamente, estamos haciendo una *decodificación de máxima verosimilitud completa (CMLD).*
- Si pedimos una retransmisión, estamos haciendo una *decodificación de máxima verosimilitud incompleta (IMLD).*

1.4. Distancia de Hamming

Supongamos que se están enviando palabras de un código C a través de un BSC con probabilidad cruzada $p < \frac{1}{2}$. Si $\mathbf{x}$ es una palabra recibida, entonces

$$\mathcal{P}(\text{ h.r. } \mathbf{x} \mid \text{h.e. } \mathbf{c}) = p^e(1-p)^{n-e} \quad (\forall \mathbf{c} \in C) \tag{1.1}$$

donde e es el número de posiciones en las que $\mathbf{x}$ difiere de $\mathbf{c}$.
Como $p < \frac{1}{2}$, la probabilidad (1.1) será máxima cuando e sea mínimo.

Definición 1.4.1. *Dado un alfabeto $\mathbb{F}_q$, definimos la función*

$$\begin{aligned} \chi : \mathbb{F}_q \times \mathbb{F}_q &\longrightarrow \mathbb{F}_2 \\ (x, y) &\rightsquigarrow \begin{cases} 1 & si\ x \neq y \\ 0 & si\ x = y \end{cases} \end{aligned}$$

y definimos la distancia de Hamming[2] *de las palabras $\mathbf{x}$, $\mathbf{y}$ (de longitud n) sobre $\mathbb{F}_q$ como*

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \chi(x_i, y_i)$$

[1] Pese a que con "palabra" nos referimos a cualquier elemento del código, por costumbre también nos refereriremos con este término a un vector.

[2] Richard Hamming (1915-1998). Matemático estadounidense que también aportó, en cuanto a la teoría de códigos se refiere, el peso de Hamming, la cota de Hamming y hasta su propio código: el código de Hamming.
(`https://en.wikipedia.org/wiki/Richard_Hamming`)

(el número de posiciones en las que $\mathbf{x}$ *difiere de* $\mathbf{y}$*)*

Ejemplo 1.4.2. Sean $\mathbf{x} = 1234, \mathbf{y} = 1423, \mathbf{z} = 3214 \in \mathbb{F}_5^4$. Entonces

$$d(\mathbf{x}, \mathbf{y}) = 3, \qquad d(\mathbf{y}, \mathbf{z}) = 4, \qquad d(\mathbf{z}, \mathbf{x}) = 2.$$

Proposición 1.4.3. *Sean* $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ *palabras (de longitud* n*) sobre* $\mathbb{F}_q$*. Entonces:*

(i) $0 \leq d(\mathbf{x}, \mathbf{y}) \leq n$

(ii) $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$

(iii) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$

(iv) $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$ *(desigualdad tirangular)*

Así que $(\mathbb{F}_q, d)$ *es un espacio métrico.*

Demostración. $\boxed{(i), (ii) \text{ y } (iii)}$ Se siguen de la definición.

$\boxed{(iv)}$ Sean x_i, y_i, z_i las posiciones i-ésimas de $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ respectivamente. Probando que

$$\chi(x_i, z_i) \leq \chi(x_i, y_i) + \chi(y_i, z_i)$$

y sumando en i quedará probado (iv).
Si $\chi(x_i, z_i) = 0$, entonces la desigualdad es cierta porque χ es semipositiva.
Si $\chi(x_i, z_i) = 1$, no puede ocurrir $x_i = y_i = z_i$, luego $\chi(x_i, y_i) = 1$ o $\chi(y_i, z_i) = 1$, dándose de nuevo la desigualdad. □

1.5. Decodificación por el vecino más cercano o por distancia mínima

Supongamos que se están enviando palabras de un código $C \subseteq \mathbb{F}_q^n$ a través de un canal de comunicación. Dada una palabra $\mathbf{x} \in \mathbb{F}_q^n$ recibida, la *regla de decodificación por el vecino más cercano* (o *regla de decodificación por distancia mínima*) establece que la palabra del código más propensa a haber sido enviada es la palabra $\mathbf{c_x} \in C$ tal que

$$d(\mathbf{x}, \mathbf{c_x}) = \min_{\mathbf{c} \in C} d(\mathbf{x}, \mathbf{c}) \tag{1.2}$$

Tal y como ocurre con la decodificación por máxima verosimilitud, de haber varias $\mathbf{c_x} \in C$ cumpliendo (1.2) podemos, o bien elegir una de ellas arbitrariamente (regla de decodificación completa) o bien pedir una retransmisión (regla de decodificación incompleta).

Teorema 1.5.1. *En cualquier BSC con probabilidad cruzada* $p < \frac{1}{2}$*, la regla de decodificación por máxima verosimilitud (MLD) es la misma que la regla de decodificación por el vecino más cercano.*

Demostración. Si C es el código que estamos usando, $\mathbf{x} \in \mathbb{F}_q^n$ es una palabra recibida y $\mathbf{c} \in C$, sabemos que

$$d(\mathbf{x}, \mathbf{c}) = i \Leftrightarrow \mathcal{P}(\text{ h.r. } \mathbf{x} \mid \text{h.e. } \mathbf{c}) = p^i(1-p)^{n-i} \quad \forall i \in \{0, \ldots, n\} \tag{1.3}$$

Como $p < \frac{1}{2}$, entonces $p < 1 - p$, luego $p(1-p)^{-1} < 1$ y

$$p^0(1-p)^n > p^1(1-p)^{n-1} > \ldots > p^n(1-p)^0$$

Por tanto, la regla de decodificación por máxima verosimilitud decodificará $\mathbf{x}$ por $\mathbf{c} \in C$ de manera que i de (1.3) sea máxima; esto es precisamente lo que hace la regla de decodificación por el vecino más cercano. □

El anterior teorema también es cierto si el canal no es binario, ya que

$$d(\mathbf{x}, \mathbf{c}) = i \Leftrightarrow \mathcal{P}(\text{ h.r. } \mathbf{x} \mid \text{h.e. } \mathbf{c}) = \left(\frac{p}{q-1}\right)^i (1-p)^{n-i} \quad \forall i \in \{0, \ldots, n\}$$

y $\frac{p}{q-1} < p < 1 - p$, luego $\frac{p}{q-1}(1-p)^{-1} < 1$ y

$$\left(\frac{p}{q-1}\right)^0 (1-p)^n > \left(\frac{p}{q-1}\right)^1 (1-p)^{n-1} > \ldots > \left(\frac{p}{q-1}\right)^n (1-p)^0$$

Observación 1.5.2. De aquí en adelante asumiremos que todos los BSC tienen probabilidades cruzadas $p < \frac{1}{2}$. Por tanto, en estos canales siempre podremos hacer la MLD mediante la decodificación por distancia mínima.

Ejemplo 1.5.3. Supongamos que se están enviando palabras del código $\{0000, 0011, 1000\}$ a través de un BSC. Si $\mathbf{x} = 0111$ es la palabra recibida, entonces

$$d(0111, 0000) = 3, \qquad d(0111, 0011) = 1, \qquad d(0111, 1000) = 4.$$

Según la decodificación por el vecino más cercano, decodificamos $\mathbf{x}$ por 0011.

1.6. Distancia mínima de un código

Definición 1.6.1. *Definimos la* distancia mínima *de un código C con $|C| \geq 2$ como*

$$d(C) = \min_{\substack{\mathbf{x},\, \mathbf{y} \in C \\ \mathbf{x} \neq \mathbf{y}}} d(\mathbf{x}, \mathbf{y})$$

Definición 1.6.2. *Llamaremos (n, M, d)-código a un (n, M)-código con distancia mínima d.*

Ejemplo 1.6.3. Sea $C = \{000000, 000111, 111222\}$ un código ternario. Entonces

$$d(000000, 000111) = 3, \qquad d(000000, 111222) = 6, \qquad d(000111, 111222) = 6;$$

por lo que $d(C) = 3$ y C es un (6, 3, 3)-código.

Definición 1.6.4. *Sea $u \in \mathbb{Z}^+$:*

Un código C es u-detector *si al modificar cualquier palabra de C entre 1 y u posiciones (ambas inclusive) se obtiene una palabra que no está en C. Con símbolos:*

$$\forall \mathbf{c} \in C \quad \forall \mathbf{x} \ \ [1 \leq d(\mathbf{c}, \mathbf{x}) \leq u \Rightarrow \mathbf{x} \notin C]$$

Un código C es exactamente u-detector *si es u-detector pero no $(u+1)$-detector.*

Ejemplo 1.6.5. Hemos visto que para $C = \{000000, 000111, 111222\}$ se tiene $d(C) = 3$, por lo que

$$\forall \mathbf{c} \in C \quad \forall \mathbf{x} \ \ [1 \leq d(\mathbf{c}, \mathbf{x}) \leq 2 \Rightarrow \mathbf{x} \notin C]$$

es decir, C es 2-detector; pero no 3-detector, ya que

$$1 \leq d(000000, 000111) \leq 3 \not\Rightarrow 000111 \notin C,$$

luego C es exactamente 2-detector

Proposición 1.6.6. *Un código C es u-detector si y sólo si $d(C) \geq u + 1$.*

Demostración. $\boxed{\Rightarrow}$ Si fuera $d(C) = s \leq u$, habría una palabra de C de manera que, modificándola en s posiciones, obtengamos otra palabra de C. Entonces C no sería u-detector.

$\boxed{\Leftarrow}$ Si no fuera u-detector, habría una palabra de C de manera que, modificándola en u posiciones, obtengamos otra palabra de C. Entonces $d(C) \leq u$. □

Teorema 1.6.7. *Todo código C es exactamente $(d(C) - 1)$-detector.*

Demostración. Es consecuencia de la proposición anterior. □

Definición 1.6.8. *Sea $v \in \mathbb{Z}^+$:*

Un código C es v-corrector *si, recibido un vector de $\mathbb{F}_q^n$ con hasta v errores (v inclusive), somos capaces de decidir qué palabra fue enviada mediante la regla de decodificación incompleta por distancia mínima.*

Un código C es exactamente v-corrector *si es v-corrector pero no $(v+1)$-corrector.*

Ejemplo 1.6.9. Sea el código binario $C = \{000, 111\}$. Usando la regla de decodificación por distancia mínima:

- Si enviamos 000 y ocurre un error (un cambio de dígito en una posición) en la transmisión, la palabra recibida se decodificará por 000.

- Si enviamos 111 y sucede lo mismo, la palabra recibida se decodificará por 111.

Por tanto, C es 1-corrector; sin embargo,

$$000 \xrightarrow{\text{2 errores}} 011 \xrightarrow{\text{decodificamos}} 111 \neq 000,$$

por lo que C no es 2-corrector. Así, C es exactamente 1-corrector.

Denotaremos con $\lfloor \cdot \rfloor$ a la parte entera de un número real dado.

Proposición 1.6.10. *Un código C es v-corrector si y sólo si $d(C) \geq 2v+1$.*

Demostración. $\boxed{\Rightarrow}$ Supongamos $d(C) \leq 2v$ y sean $\mathbf{c}, \mathbf{c}' \in C$. Si $d(\mathbf{c}, \mathbf{c}') \leq v$, podríamos obtener $\mathbf{c}'$ de $\mathbf{c}$ modificando v o menos posiciones, por lo que C no sería v-corrector. Debe ser entonces $d(\mathbf{c}, \mathbf{c}') \geq v+1$. Supongamos sin pérdida de generalidad que $\mathbf{c}$ y $\mathbf{c}'$ difieren en las $d = d(C)$ primeras posiciones con $v+1 \leq d \leq 2v$ y sea

$$\mathbf{x} = x_1 \dots x_v x_{v+1} \dots x_d x_{d+1} \dots x_k \text{ con } x_i = \begin{cases} c_i & (i = 1, \dots, v) \\ c'_i & (i = v+1, \dots, d) \\ c_i = c'_i & (i = d+1, \dots, k) \end{cases}$$

una palabra recibida. Entonces $d(\mathbf{x}, \mathbf{c}') = d - v \leq v = d(\mathbf{x}, \mathbf{c})$:
Si fuera $d(\mathbf{x}, \mathbf{c}') < d(\mathbf{x}, \mathbf{c})$, concluiríamos que $\mathbf{c}'$ fue la palabra enviada, pero como $d(\mathbf{x}, \mathbf{c}) = v$, podríamos afirmar que C no es v-corrector.
Si fuera $d(\mathbf{x}, \mathbf{c}') = d(\mathbf{x}, \mathbf{c}) = v$, tendríamos un empate, luego C no sería v-corrector.

$\boxed{\Leftarrow}$ Sea $\mathbf{x}$ una palabra recibida de forma que $d(\mathbf{x}, \mathbf{c}) \leq v$ para cierto $\mathbf{c} \in C$. No puede existir ninguna otra palabra $\mathbf{c}' \in C$ de forma que $d(\mathbf{x}, \mathbf{c}') \leq v$, ya que de lo contrario $d(\mathbf{c}, \mathbf{c}') \leq 2v$. Concluimos entonces que $\mathbf{c}$ fue la palabra enviada, luego C es v-corrector. □

Teorema 1.6.11. *Todo código C es exactamente* $\left\lfloor \frac{d(C)-1}{2} \right\rfloor$*-corrector.*

Demostración. Es consecuencia de la proposición anterior. □

Bibliografía del capítulo

Asensio-Caruncho-Martínez [1], Ling-Xing [6].

Capítulo 2

Códigos lineales

Entre las familias de códigos más antiguas se encuentra la de los códigos lineales. Con estos códigos, la idea es añadir redundancia mediante una aplicación lineal e inyectiva $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$. Así, $C = \mathcal{C}(\mathbb{F}_q^k)$ es un subespacio vectorial de $\mathbb{F}_q^n$. Obviamos el "paso" de $Pal(\mathcal{A})$ a $\mathbb{F}_q^k$ denotando con $\mathcal{C}$ a la aplicación lineal.

2.1. Códigos lineales

Definición 2.1.1. *Un* código lineal *C en $\mathbb{F}_q^n$ (o de longitud n sobre $\mathbb{F}_q$) es un subespacio vectorial de $\mathbb{F}_q^n$.*

Ejemplos 2.1.2.

(i) $\{(\lambda, \ldots, \lambda) : \lambda \in \mathbb{F}_q\}$ es un código lineal de longitud n sobre $\mathbb{F}_q$.

(ii) $\{0000, 1010, 0101, 1111\}$ es un código lineal de longitud 4 sobre $\mathbb{F}_2$.

(iii) $\{000, 012, 021\}$ es un código lineal de longitud 3 sobre $\mathbb{F}_3$.

Definición 2.1.3. *Sea C un código lineal en $\mathbb{F}_q^n$.*

(i) Llamamos código dual *de C a $C^\perp = \left\{\mathbf{v} \in \mathbb{F}_q^n : \mathbf{v}\cdot\mathbf{c} = 0 \ (\forall \mathbf{c} \in C)\right\}$, donde '$\cdot$' denota el producto escalar usual.*

(ii) Llamamos dimensión *de C a la dimensión de C como $\mathbb{F}_q$-espacio vectorial. Lo denotamos como $dim_{\mathbb{F}_q}(C)$.*

Teorema 2.1.4. *Sea C un código lineal en $\mathbb{F}_q^n$. Entonces:*

(i) $C^\perp$ es un código lineal y $dim(C) + dim(C^\perp) = n$

(ii) $(C^\perp)^\perp = C$.

Demostración. $\boxed{(i)}$ De las propiedades del producto escalar se sigue que

$$\lambda\mathbf{v}_1 + \mu\mathbf{v}_2 \in C^\perp \quad \forall \mathbf{v}_1, \mathbf{v}_2 \in C^\perp,$$

luego $C^\perp$ es lineal. Si $dim(C) = k \geq 1$, $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ es una base de C y

$$G = \begin{pmatrix} — & \mathbf{c}_1 & — \\ & \vdots & \\ — & \mathbf{c}_k & — \end{pmatrix},$$

entonces el núcleo de la aplicación lineal suprayectiva

$$\begin{array}{rcl} \mathbb{F}_q^n & \to & Col(G) \\ (v_1, \ldots, v_n) & \rightsquigarrow & (v_1, \ldots, v_n)G^T \end{array}$$

(donde $Col(G)$ es el espacio generado por las columnas de G) es $C^\perp$ y, por el teorema de isomorfía, se tiene

$$\frac{\mathbb{F}_q^n}{C^\perp} \cong Col(G)$$

Entonces

$$dim\left(\frac{\mathbb{F}_q^n}{C^\perp}\right) = dim\left(Col(G)\right) = rang(G) = dim(C)$$

Ahora bien, sabemos que

$$dim\left(\frac{V}{U}\right) = dim(V) - dim(U)$$

para cualquier espacio vectorial V de dimensión finita y cualquier subespacio vectorial $U \subseteq V$. En particular es

$$dim(C) = dim\left(\frac{\mathbb{F}_q^n}{C^\perp}\right) = n - dim(C^\perp)$$

y se sigue (i).

$\boxed{(ii)}$ Si $\mathbf{c} \in C$, entonces $\mathbf{c}\cdot\mathbf{x}=0$ para todo $\mathbf{x} \in C^\perp$, por lo que $\mathbf{c} \in (C^\perp)^\perp$ y $C \subseteq (C^\perp)^\perp$. De (i) se tiene

$$dim(C) + dim(C^\perp) = n = dim(C^\perp) + dim\left((C^\perp)^\perp\right),$$

de donde se deduce la igualdad. □

Observación 2.1.5. Sustituyendo C por $C^\perp$ en la demostración anterior se tiene que $dim(C^\perp) = dim\left(\frac{\mathbb{F}_q^n}{C}\right)$, luego $C^\perp \cong \frac{\mathbb{F}_q^n}{C}$. Recordaremos esto más adelante.

Un código lineal C de longitud n y dimensión k sobre $\mathbb{F}_q$ es llamado también un $[n, k]$-código lineal q-ario. De hecho, es un (n, q^k)-código lineal. Si añadimos además la distancia $d = d(C)$ de C, podemos usar las nomenclaturas

$$(n, q^k, d)\text{-código lineal} \qquad \text{o bien} \qquad [n, k, d]\text{-código lineal}$$

2.2. Peso de Hamming

Definición 2.2.1. *Definimos el* peso *de* $\mathbf{x} \in \mathbb{F}_q^n$ *como* $wt(\mathbf{x}) = d(\mathbf{x}, \mathbf{0})$*, es decir, como el número de coordenadas no nulas de* $\mathbf{x}$*.*

Definición 2.2.2. *Definimos el* peso mínimo *de un código* C *como*

$$wt(C) = \min_{\mathbf{0} \neq \mathbf{x} \in C} wt(\mathbf{x})$$

El peso de Hamming de un código siempre existe porque los códigos son conjuntos finitos.

Teorema 2.2.3. *Todos los códigos lineales sobre* $\mathbb{F}_q$ *satisfacen* $d(C) = wt(C)$*.*

Demostración. $\boxed{\leq}$ Sea $\mathbf{0} \neq \mathbf{z} \in C$ tal que $wt(C) = wt(\mathbf{z})$. Entonces

$$wt(C) = wt(\mathbf{z}) = d(\mathbf{z}, \mathbf{0}) \geq d(C).$$

$\boxed{\geq}$ Sean $\mathbf{x}, \mathbf{y} \in C$ tales que $d(C) = d(\mathbf{x}, \mathbf{y})$. Entonces $\mathbf{x} - \mathbf{y} \in C$ por ser C lineal y se tiene

$$d(C) = d(\mathbf{x}, \mathbf{y}) = wt(\mathbf{x} - \mathbf{y}) \geq wt(C)$$

□

Observaciones 2.2.4. Ventajas de los códigos lineales:

(*i*) Pueden ser descritos mediante una base (pues son $\mathbb{F}_q$-espacios vectoriales).

(*ii*) $d(C) = wt(C)$.

(*iii*) Los procesos de codificación y decodificación son, en general, más rápidos que los que hay para códigos no lineales.

2.3. Bases para códigos lineales

Denotaremos con $RREF(A)$ a la matriz escalonada reducida por filas que se obtiene de $A \in \mathbb{M}(\mathbb{F}_q)$ mediante operaciones elementales fila y llamaremos *columnas líder* a las columnas de $RREF(A)$ donde se encuentran los pivotes. Recordemos que toda matriz $A \in \mathbb{M}(\mathbb{F}_q)$ es equivalente por filas a una única matriz escalonada reducida por filas.

Mediante el siguiente algoritmo, que justificaremos más adelante, podemos construir bases de códigos duales.

Entrada: Subcojunto no vacío S de $\mathbb{F}_q^n$.

Salida: Una base del código dual $C^\perp$ del código lineal C generado por S, es decir: una base de $C^\perp$, donde $C = \langle S \rangle$

Descripción:

$$A = \left(\begin{array}{c} \text{———} \\ \text{— } S \text{ —} \\ \text{———} \end{array}\right) \quad \xrightarrow{\text{Operaciones elem. fila}} \dots \xrightarrow{} \quad RREF(A) = \left(\begin{array}{c} G \\ \hline 0 \end{array}\right)$$

Si llamamos k al número de filas de G (el número de columnas debe ser n, pues estamos en $\mathbb{F}_q^n$), entonces G debe tener k columnas líder, por lo que

$$G \xrightarrow{\substack{\text{Permutar} \\ \text{columnas}}} G' = (I_k|X) \to H' = (-X^T|I_{n-k}) \xrightarrow{\substack{\text{Invertir la} \\ \text{permutación}}} H = \left(\begin{array}{c} \text{———} \\ \text{base de } C^\perp \\ \text{———} \end{array}\right)$$

Ejemplo 2.3.1. En $\mathbb{F}_3^{10}$ y dada

$$G = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 2 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix},$$

hallar una base de $C^\perp$. Aplicando la permutación $\sigma = (2\ 4\ 7\ 3\ 5\ 9\ 8\ 6)$ sobre las columnas de G, tenemos

$$G' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\downarrow$$

$$H' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\downarrow$$

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 1 & 1 \end{pmatrix}$$

Las filas de H forman una base de $C^\perp$.

2.4. Matriz generadora y matriz de control de paridad

Definición 2.4.1.

(i) Una matriz generadora *del código lineal C es una matriz* $G = \begin{pmatrix} \overline{\qquad\qquad} \\ \text{base de } C \\ \overline{\qquad\qquad} \end{pmatrix}$

(ii) Una matriz de control de paridad *(o* matriz de control*) del código lineal C es una matriz generadora H de $C^\perp$, es decir, una matriz* $H = \begin{pmatrix} \overline{\qquad\qquad} \\ \text{base de } C^\perp \\ \overline{\qquad\qquad} \end{pmatrix}$

Observaciones 2.4.2.

(i) Si C es un $[n,k]$-código lineal, entonces

- la matriz generadora G de C es una matriz $k \times n$.
- la matriz de control de paridad H de C es una matriz $(n-k) \times n$ por tener $C^\perp$ dimensión $n-k$ por el Teorema 2.1.4.

(ii) En general, los espacios vectoriales tienen más de una base. Esto también ocurre con el número de matrices generadoras de un código lineal. De hecho, cualquier permutación de filas de una matriz generadora de C da lugar a una nueva matriz generadora de C.

Definición 2.4.3.

(i) Diremos que una matriz generadora G está en la forma estándar *si tiene la forma* $(I_k|X)$.

(ii) Diremos que una matriz de control de paridad H está en la forma estándar *si tiene la forma* $(Y|I_{n-k})$.

Lema 2.4.4. *Sea C un $[n,k]$-código lineal sobre $\mathbb{F}_q$ con matriz generadora G. Entonces*

$$\forall \mathbf{v} \in \mathbb{F}_q^n \quad \left[\mathbf{v} \in C^\perp \Leftrightarrow \mathbf{v}G^T = \mathbf{0}\right]$$

En particular, una matriz $H \in \mathbb{M}_{(n-k)\times n}(\mathbb{F}_q^n)$ es de control si sólo si sus filas son linealmente independientes y $HG^T = 0$.

Teorema 2.4.5. *Sea C un código lineal con matriz de control de paridad H:*

(i) $d(C) \geq d \Leftrightarrow$ Cualesquiera $d-1$ columnas de H son linealmente independientes.

(ii) $d(C) \leq d \Leftrightarrow H$ tiene d columnas linealmente dependientes.

Demostración. $\boxed{(i)}$ $\Rightarrow$: Supongamos sin pérdida de generalidad que las primeras $d-1$ columnas $c_1, \ldots, c_{d-1}$ de H son linealmente dependientes. Entonces existen $\lambda_1, \ldots, \lambda_{d-1} \in \mathbb{F}_q$ no todos nulos tales que

$$\lambda_1 c_1 + \ldots + \lambda_{d-1} c_{d-1} = 0 \tag{2.1}$$

Sea $\mathbf{v} = (\lambda_1, \ldots, \lambda_{d-1}, 0, \ldots, 0) \in \mathbb{F}_q^n$. Entonces $H\mathbf{v}^T = \mathbf{0}^T$ por (2.1), luego $\mathbf{v} \in C$ y

$$d(C) = wt(C) \leq wt(\mathbf{v}) = d - 1 < d$$

$\Leftarrow$: Si fuera $d(C) < d$, existiría $\mathbf{v} \in C$ tal que $wt(\mathbf{v}) \leq d - 1$. Sin pérdida de generalidad, supongamos que las coordenadas no nulas de $\mathbf{v}$ son las $d-1$ primeras $v_1, \ldots, v_{d-1}$. Entonces

$$\mathbf{0}^T = H\mathbf{v}^T = v_1c_1 + \ldots + v_{d-1}c_{d-1}$$

donde $c_1, \ldots, c_{d-1}$ son las $d-1$ primeras columnas de H. Hemos encontrado pues $d-1$ columnas linealmente dependientes.

$\boxed{(ii)}$ $\Rightarrow$: Razonando igual que en "$\Leftarrow$"del apartado anterior con $d(C), wt(\mathbf{v}) \leq d$ y las d primeras coordenadas $v_1, \ldots, v_d$ se tiene la implicación.

$\Leftarrow$: Razonando igual que en "$\Rightarrow$"del apartado anterior con las d primeras columnas $c_1, \ldots, c_d$ de H obtenemos que $d(C) = wt(C) \leq wt(\mathbf{v}) = d$. □

Corolario 2.4.6. *Sea C un código lineal con matriz de control de paridad H. Son equivalentes:*

(i) $d(C) = d$

(ii) Cualesquiera $d-1$ columnas de H son linealmente independientes y H tiene d columnas linealmente dependientes.

Como la matriz de control tiene rango $n - k$, entonces cualesquiera $n - k + 1$ columnas de H son linealmente independientes. Por el anterior teorema tenemos la llamada *cota de Singleton*: $d \leq n - k + 1$.

Ejemplo 2.4.7. Sea C un código lineal en $\mathbb{F}_2^5$ con matriz de control de paridad

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

Las columnas 1, 3 y 4 suman $\mathbf{0}^T$, luego son linealmente dependientes. Nos preguntamos si todo par de columnas de H es linealmente independiente. Como el cuerpo base es $\mathbb{F}_2$, esto es equivalente a preguntarse si la suma de cualesquiera dos columnas de H es distinta de $\mathbf{0}^T$. La respuesta es afirmativa, luego $d(C) = 3$.

Teorema 2.4.8. *Sea C un $[n, k]$-código lineal con matriz generadora en su forma estándar $G = (I_k|X)$. Entonces $H = (-X^T|I_{n-k})$ es una matriz de control de C.*

Demostración. Las filas de H son linealmente independientes. Como también $HG^T = 0$, por el Lema 2.4.4 H es una matriz de control. □

Ejemplo 2.4.9. Dado $S = \{11101, 10110, 01011, 11010\} \subseteq \mathbb{F}_2^5$, encontrar una matriz generadora G y una matriz de control H del código lineal $C = \langle S \rangle$. Aplicando el algoritmo visto obtenemos

$$G' = G = \left(\begin{array}{c|cc} & 0 & 1 \\ I_3 & 1 & 1 \\ & 1 & 1 \end{array}\right) \to H = H' = \left(\begin{array}{ccc|c} 0 & 1 & 1 & \\ 1 & 1 & 1 & I_2 \end{array}\right) = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

H es la matriz de control de paridad de C.

Ejemplo 2.4.10. Si bien en general un código lineal tiene más de una matriz generadora, también pude ocurrir que no tenga ninguna en forma estándar.

Sea el código lineal $C = \{000, 001, 100, 101\} \subseteq \mathbb{F}_2^3$. Entonces C tiene dimensión 2 y $\frac{1}{2!}(2^2 - 2^0)(2^2 - 2^1) = 3$ bases, que son

$$\{001, 100\}, \qquad \{001, 101\} \quad \text{y} \quad \{100, 101\}.$$

Por tanto, obtenemos seis matrices generadoras, ninguna en forma estándar:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

2.5. Equivalencia de códigos lineales

Definición 2.5.1. *Dos (n, M)-códigos sobre $\mathbb{F}_q$ son* equivalentes *si uno puede obtenerse del otro mediante*

(i) permutación de n dígitos de las palabras y/o

(ii) multiplicación de los dígitos de una posición fijada por un escalar no nulo.

Ejemplos 2.5.2.

(i) Dado $C = \{0000, 0101, 0010, 0111\} \subseteq \mathbb{F}_2^4$, aplicando la permutación $\sigma = (1\ 2\ 4\ 3)$ a los dígitos de C, obtenemos el código equivalente $C' = \{0000, 1100, 0001, 1101\}$.

(ii) Dado $C = \{000, 011, 022\} \subseteq \mathbb{F}_3^3$, aplicando la trasposición $\sigma = (1\ 2)$ a los dígitos de C y multiplicando por dos el último dígito, obtenemos el código equivalente $C' = \{000, 102, 201\}$.

De la definición anterior se deduce que dos códigos equivalentes tienen los mismos parámetros (longitud, dimensión y distancia[1]), pero recordemos (Ejemplo 2.4.10) que no todo

[1] Así que dos códigos equivalentes son isométricos.
(https://en.wikipedia.org/wiki/Isometry#Formal_definitions)

código lineal tiene por qué tener una matriz generadora en forma estándar. Se tiene el sguiente teorema:

Teorema 2.5.3. *Todo código lineal C es equivalente a otro también lineal C' con matriz generadora en forma estándar.*

Observación 2.5.4. Permutar las columnas de una matriz generadora G de un código lineal C es entonces hallar matrices generadoras de códigos lineales equivalentes a C. En el algoritmo 2.3 precisamente hallábamos la matriz generadora en forma estándar de un código lineal equivalente al dado. De esta matriz obteníamos una matriz H' que es de control de tal código equivalente por el Teorema 2.4.8. Deshaciendo la permutación de las columnas obteníamos una matriz de control del código "original". Esto justifica la validez del algoritmo visto.

Ejemplo 2.5.5. Vimos en el Ejemplo 2.4.10 que el código lineal $C = \{000, 001, 100, 101\} \subseteq \mathbb{F}_2^3$ no tiene matrices generadoras en forma estándar. Aplicando la trasposición $\sigma = (2\ 3)$ a los dígitos de C, obtenemos el código equivalente $C' = \{000, 010, 100, 110\}$, que sigue teniendo dimensión 4 y 3 bases, una de las cuales es $\{010, 100\}$, luego $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ es una matriz generadora de C' y además está en forma estándar.

2.6. Codificando con un código lineal

Comentamos al comienzo del capítulo que añadiremos la redundancia mediante una aplicación lineal e inyectiva $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$. Para ello necesitamos una matriz asociada. Sea C un $[n, k]$-código lineal sobre $\mathbb{F}_q$. Por definición, C es un espacio vectorial, luego existe base $\{\mathbf{r}_1, \ldots, \mathbf{r}_k\}$ de C y

$$G = \begin{pmatrix} — & \mathbf{r}_1 & — \\ & \vdots & \\ — & \mathbf{r}_k & — \end{pmatrix}$$

es una matriz generadora de C. Podemos así considerar la aplicación lineal e inyectiva $\mathcal{C}_G : \mathbb{F}_q^k \to C \leq \mathbb{F}_q^n$, donde cada $\mathbf{u} \in \mathbb{F}_q^k$ se codifica como $\mathbf{v} = \mathbf{u}G \in C$.

Ejemplo 2.6.1. Sea C un $[5, 3]$-código lineal sobre $\mathbb{F}_2$ con matriz generadora

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

Entonces el mensaje $\mathbf{u} = 101 \in \mathbb{F}_2^3$ es codificado como

$$\mathbf{v} = \mathbf{u}G = (101)\begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} = 10011$$

La tasa de información de C es $\frac{3}{5}$, es decir, se usan 3 símbolos de 5 para transmitir el mensaje.

Observaciones 2.6.2. Ventajas de tener una matriz generadora en su forma estándar $G = (I|X)$:

(i) Que también tenemos la matriz de control de paridad $H = (-X^T|I)$

(ii) Que podemos recuperar fácilmente el mensaje $\mathbf{u}$ de la palabra del código $\mathbf{v}$, pues

$$\mathbf{v} = \mathbf{u}G = \mathbf{u}(I|X) = (\mathbf{u}, \mathbf{u}X)$$

(las k primeras posiciones de $\mathbf{v}$ son las del mensaje $\mathbf{u}$)

Dado un código lineal C con matriz generadora G, se dice que k columnas de G forman un *conjunto de información* si son linealmente independientes. En tal caso, se dice que las restantes $n - k$ columnas forman un *conjunto de redundancia*, y si consideramos el producto $\mathbf{u}G$, a las correspondientes posiciones de las k columnas se les llama *posiciones de información* y al resto *posiciones de control*. Como toda matriz sobre $\mathbb{F}_q$ es equivalente por filas a una RREF, toda matriz generadora tiene al menos un conjunto de información y conjunto de redundancia. Si en particular la matriz generadora está en forma estándar, la ventaja es clara: las k primeras posiciones son posiciones de información.

2.7. Decodificación de los códigos lineales

2.7.1. Espacios cociente

Sea C un código lineal en $\mathbb{F}_q^n$ y considerémos el espacio cociente $\mathbb{F}_q^n/C$ formado por las clases laterales. Dado $\mathbf{u} \in \mathbb{F}_q^n$, denotaremos, como es usual, la clase lateral de C determinada por $\mathbf{u}$ como

$$C + \mathbf{u} = \{\mathbf{v} + \mathbf{u} \ : \ \mathbf{v} + C\} \ \ (= \mathbf{u} + C)$$

Ejemplo 2.7.1. Sea $C = \{000, 101, 010, 111\} \subseteq \mathbb{F}_2^3$. Entonces

$$C + 000 = \{000, 101, 010, 111\} \qquad C + 001 = \{001, 100, 011, 110\}$$
$$C + 010 = \{010, 111, 000, 101\} \qquad C + 011 = \{011, 110, 001, 100\}$$
$$\ldots$$

Se tiene que

$$\begin{aligned} C + 000 = C + 010 = C + 101 = C + 111 &= C \\ C + 001 = C + 011 = C + 100 = C + 110 &= \mathbb{F}_2^3 \setminus C \end{aligned}$$

Definición 2.7.2. *A la palabra de menor peso de una clase se le llama* líder *(o* palabra líder*) de la clase.*

Ejemplo 2.7.3. El líder de una clase no tiene por qué ser único. Para el código del Ejemplo 2.7.1 $C = \{000, 101, 010, 111\} \subseteq \mathbb{F}_2^3$ tenemos las clases

$$C + 000 = \{000, 101, 010, 111\} \qquad \text{y} \qquad C + 001 = \{001, 100, 011, 110\}$$

La primera clase tiene la palabra líder 000, mientras que la segunda clase tiene las palabras líder 001 y 100.

Teorema 2.7.4. *Sea C un código lineal. Si $\mathbf{e} \in \mathbb{F}_q^n$ satisface $wt(\mathbf{e}) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, entonces es la única palabra de $\mathbf{e} + C$ con peso mínimo.*

Demostración. Si existiera otra palabra $\mathbf{e}' = \mathbf{e} + \mathbf{c}$ de $\mathbf{e} + C$ con $wt(\mathbf{e}') \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, se tendría

$$wt(\mathbf{c}) = wt(\mathbf{e}' - \mathbf{e}) \leq wt(\mathbf{e}') + wt(\mathbf{e}) \leq 2 \left\lfloor \frac{d(C)-1}{2} \right\rfloor \leq d(C) - 1 = wt(C) - 1,$$

lo cual implica $\mathbf{c} = \mathbf{0}$ y $\mathbf{e}' = \mathbf{e}$. □

2.7.2. Decodificación por el vecino más cercano para códigos lineales

Sea C un código lineal y supongamos que se envía una palabra del código $\mathbf{v}$ y que se recibe una palabra $\mathbf{w}$, dando lugar al *patrón de error* (o *cadena de error*) $\mathbf{e} = \mathbf{w} - \mathbf{v}$. Entonces $\mathbf{w} - \mathbf{e} = \mathbf{v} \in C$, por lo que $\mathbf{w}$ y $\mathbf{e}$ están en la misma clase lateral.

Si establecemos como hipótesis que no se cometen más errores que la capacidad de corrección, por el Teorema 2.7.4 existe un único líder $\mathbf{e}$ en la clase lateral $\mathbf{w} + C$ y diremos entonces que la palabra enviada fue $\mathbf{w} - \mathbf{e}$. Si se cometen más errores, o bien pedimos una retransmisión, o bien decodificamos arbitrariamente por una palabra de $\mathbf{w} + C$, pudiendo tomar una decisión equivocada en este último caso.

Ejemplo 2.7.5. Sea $C = \{\mathbf{0}, 11100, 01111, 10011\} \subseteq \mathbb{F}_2^5$. Vamos a decodificar las palabras

$$(i)\ \mathbf{w}_1 = 01011 \qquad \text{y} \qquad (ii)\ \mathbf{w}_2 = 10110.$$

Hay $2^{5-2} = 8$ clases laterales y son:

	$00000+C:$	00000	11100	01111	10011
	$10000+C:$	10000	01100	11111	00011
	$01000+C:$	01000	10100	00111	11011
	$00100+C:$	00100	11000	01011	10111
	$00010+C:$	00010	11110	01101	10001
	$00001+C:$	00001	11101	01110	10010
$*$	$00110+C:$	00110	11010	01001	10101
$*$	$01010+C:$	01010	10110	00101	11001

La tabla está escrita de tal forma que en la primera columna están las palabras líder de cada clase. En * hay dos palabras líder; hemos elegido 00110 y 01010 arbitrariamente. Como tomaremos como patrón de error $\mathbf{e}$ la palabra de menor peso de la clase de $\mathbf{w}_i$ (es decir, la palabra líder de la clase de $\mathbf{w}_i$), resulta que decodificar por $\mathbf{w}_i - \mathbf{e}$ es lo mismo que decodificar por la palabra que está en la primera fila y en la misma columna que $\mathbf{w}_i$.

(i) $\mathbf{w}_1 = 01011$: Vemos que está en la tercera clase. Entonces el patrón de error (la palabra líder) es $\mathbf{e} = 00100$ y decodificamos por

$$\mathbf{w}_1 - \mathbf{e} = 01011 - 00100 = 01111$$

(ii) $\mathbf{w}_2 = 10110$: Vemos que está en la última clase. Si estamos haciendo la decodificación incompleta, pedimos una retransmisión (pues hay "empate" para ser $\mathbf{e}$). Si estamos haciendo la decodificación completa, como escogimos 01010 como palabra líder (patrón de error), entonces el patrón de error (palabra líder) es $\mathbf{e} = 01010$ y decodificamos por

$$\mathbf{w}_2 - \mathbf{e} = 10110 - 01010 = 11100$$

Volvemos a decir que podemos estar tomando una decisión equivocada al estar fuera de las hipótesis sobre el número de errores.

2.7.3. Decodificación por síndrome

Definición 2.7.6. *Sea C un $[n,k]$-código lineal sobre $\mathbb{F}_q$ con matriz de control de paridad H. Definimos la aplicación* síndrome S_H *(o S) como*

$$\begin{array}{rccl} S_H(\cdot): \mathbb{F}_q^n & \to & \mathbb{F}_q^{n-k} \\ \mathbf{w} & \rightsquigarrow & \mathbf{w}H^T \ (\text{síndrome } \textit{de } \mathbf{w}) \end{array}$$

Teorema 2.7.7. *Sea C un $[n,k]$-código lineal con matriz de control de paridad H. Para cualesquiera $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$ se tiene:*

(i) $S(\mathbf{u}+\mathbf{v}) = S(\mathbf{u}) + S(\mathbf{v})$

(ii) $S(\mathbf{u}) = \mathbf{0} \Leftrightarrow \mathbf{u} \in C$

(iii) $S(\mathbf{u}) = S(\mathbf{v}) \Leftrightarrow \mathbf{u}$ *y* $\mathbf{v}$ *están en la misma clase lateral de C.*

Demostración. $\boxed{(i)}$ Se sigue de la definición.

$\boxed{(ii)}$ $S(\mathbf{u}) = \mathbf{0} \Leftrightarrow \mathbf{u}H^T = \mathbf{0} \Leftrightarrow \mathbf{u} \in C$ por el Lema 2.4.4.

$\boxed{(iii)}$ $S(\mathbf{u}) = S(\mathbf{v}) \Leftrightarrow S(\mathbf{u} - \mathbf{v}) = \mathbf{0} \Leftrightarrow \mathbf{u} - \mathbf{v} \in C \Leftrightarrow \mathbf{u}$ y $\mathbf{v}$ están en la misma clase lateral. □

Observaciones 2.7.8.

(i) De (iii) del teorema anterior se deduce que hay una correspondencia biyectiva "clase − síndrome"

$$\begin{array}{rcl} \{\text{clases de } C\} & \leftrightarrow & Im(S) \\ \mathbf{u} + C & \rightsquigarrow & S(\mathbf{u}) \end{array}$$

(ii) Por el Teorema 2.7.7 (ii) es $Ker(S) = C$ y por la Observación 2.1.5 y el teorema de isomorfía se tiene

$$C^{\perp} \cong \frac{\mathbb{F}_q^n}{C} \cong Im(S),$$

así que $dim\,(Im(S)) = dim(C^{\perp}) = n - k$ por el Teorema 2.1.4 (i) y S es suprayectiva. Podemos entonces ver $\mathbb{F}_q^{n-k}$ como un conjunto de síndromes distintos.

Definición 2.7.9. *Se llama* tabla de síndromes *a cualquier tabla que muestre cada líder de clase con su síndrome.*

Pasos para construir una tabla de síndromes en términos de la decodificación completa por el vecino más cercano

Paso 1: Encontrar todas las clases del código y las palabras líder de éstas. Si en alguna clase hay más de una palabra líder, elegir una arbitrariamente).

Paso 2: Dar una matriz de control de paridad H para el código y calcular $S(\mathbf{u}) = \mathbf{u}H^T$ para cada una de las palabras líder buscadas y elegidas $\mathbf{u}$.

Ejemplo 2.7.10. Usando la decodificación completa por el vecino más cercano, construir una tabla de síndromes para el código $C = \{\mathbf{0}, 11100, 01111, 10011\} \subseteq \mathbb{F}_2^5$.

Es el código del Ejemplo 2.7.5. Tenemos entonces calculadas las clases de C y buscada y elegida la palabra líder $\mathbf{u}_i$ de cada clase. Siguiendo el algoritmo que comentamos, una matriz de control de C es

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

Calculamos los síndromes de las palabras líderes:

$$\begin{pmatrix} — & \mathbf{u}_1 & — \\ & \vdots & \\ — & \mathbf{u}_8 & — \end{pmatrix} H^T = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Ya podemos construir la tabla:

Palabra líder $\mathbf{u}$	Síndrome $S(\mathbf{u})$
00000	000
10000	011
01000	111
00100	100
00010	010
00001	001
* 00110	110
* 01010	101

El asterisco indica que, de tener síndrome 110 o 101 la palabra recibida, tendríamos que pedir una retransmisión al haber varias palabras líder en esas clases. Esto siempre y cuando estemos usando la decodificación incompleta por el vecino más cercano.

Observación 2.7.11. Recordemos que

$$\forall \mathbf{e} \in \mathbb{F}_q^n, \quad wt(\mathbf{e}) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor \Rightarrow \mathbf{e} \text{ es la única palabra líder de } \mathbf{e} + C,$$

por lo que podemos construir la tabla de síndromes más rápido si buscamos tales $\mathbf{e}$ y calculamos su síndrome $S(\mathbf{e})$. Si tras esto hubiera algún síndrome $S(\mathbf{u})$ sin correspondiente palabra lider $\mathbf{u}$ de peso $\leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$ en la tabla, tal clase de la palabra que falta podría tener más de un candidato a palabra líder. Esto es justamente lo que ocurre con los síndromes 110 y 101 del ejemplo anterior.

Ejemplo 2.7.12. Usando la decodificación completa por el vecino más cercano, construir la tabla de síndromes para el código lineal C sobre $\mathbb{F}_2$ con matriz de control de paridad

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

Calculemos $d(C)$ a partir del Corolario 2.4.6. Las columnas 2, 3 y 4 suman $\mathbf{0}^T$, luego son linealmente dependientes. Nos preguntamos si todo par de columnas de H es linealmente

independiente. Como el cuerpo sobre el que está C es $\mathbb{F}_2$, esto es equivalente a preguntarse si la suma de cualesquiera dos columnas de H es distinta de $\mathbf{0}^T$. La respuesta es afirmativa, luego $d(C) = 3$.

Como H tiene $n - k = 3$ filas, entonces hay $q^{n-k} = 2^3 = 8$ clases de C distintas. Los vectores $\mathbf{e}_1, \ldots, \mathbf{e}_6$ de la base canónica de $\mathbb{F}_2^6$ tienen peso $1 \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor = 1$, luego son las únicas palabras que son líderes de sus clases $\mathbf{e}_i + C$. Tenemos pues siete clases:

$$\mathbf{0} + C \qquad \mathbf{e}_1 + C \qquad \ldots \qquad \mathbf{e}_6 + C$$

Calculamos sus síndromes:

$$\begin{pmatrix} — & \mathbf{0} & — \\ — & \mathbf{e}_1 & — \\ & \vdots & \\ — & \mathbf{e}_6 & — \end{pmatrix} H^T = \begin{pmatrix} — & \mathbf{0} & — \\ & & \\ & H^T & \\ & & \end{pmatrix}$$

Como la aplicación síndrome es suprayectiva, deducimos que el síndrome de la palabra líder que falta es el elemento de $\mathbb{F}_2^3$ que no está en $\begin{pmatrix} - & \mathbf{0} & - \\ & H^T & \end{pmatrix}$: 101.

Buscamos entonces $\mathbf{v} \in \mathbb{F}_2^6$ tal que $\mathbf{v}H^T = 101$. Y además de menor peso, ya que, concretando, buscamos una palabra líder para poder completar al tabla. De la ecuación $\mathbf{v}H^T = 101$ se tiene que

$$\begin{array}{cccccccc} v_1 & & +v_3 & +v_4 & & & = & 1 \\ v_1 & +v_2 & +v_3 & & +v_5 & & = & 0 \\ & +v_2 & +v_3 & & & +v_6 & = & 1 \end{array}$$

La primera ecuación se satisface si y sólo si la terna (v_1, v_3, v_4) es cualquiera de las siguientes:

$$(1,0,0) \qquad (0,1,0) \qquad (0,0,1) \qquad (1,1,1)$$

Quedándonos con la primera surgen dos soluciones para el par (v_2, v_5):

$$(1,0) \qquad\qquad (0,1)$$

Quedándonos de nuevo con la primera debe ser $v_6 = 0$ y obtenemos una solución: 110000. Analizando el resto de casos y descartando los vectores con mayor peso que 110000, obtenemos otras dos soluciones: 001010 y 000101. Como estamos usando la decodificación completa, elegimos arbitrariamente 000101 y ya podemos construir la tabla:

Palabra líder $\mathbf{u}$	Síndrome $S(\mathbf{u})$
000000	000
100000	110
010000	011
001000	111
000100	100
000010	010
000001	001
* 000101	101

Si usáramos la decodificación incompleta, tendríamos que pedir una retransmisión en el caso de recibir una palabra con síndrome 101.

Proceso de decodificación para la decodificación por síndrome

Paso 1: Dada la palabra recibida $\mathbf{w}$, calcular $S(\mathbf{w})$.

Paso 2: Buscar en la tabla de decodificación por síndrome aquella palabra líder $\mathbf{u}$ tal que $S(\mathbf{w}) = S(\mathbf{u})$. Tal palabra $\mathbf{u}$ existe porque $\mathbf{w}$ debe estar en una clase y esa clase debe tener una palabra líder.

Paso 3: Decodificar $\mathbf{w}$ por $\mathbf{v} = \mathbf{w} - \mathbf{u}$.

Ejemplo 2.7.13. Sea $C = \{\mathbf{0}, 11100, 01111, 10011\} \subseteq \mathbb{F}_2^5$. Usar la tabla de decodificación por síndrome para decodificar

$$(i)\ \mathbf{w}_1 = 01011 \qquad \text{y} \qquad (ii)\ \mathbf{w}_2 = 10110.$$

Ya calculamos la tabla en el Ejemplo 2.7.10. Es

Palabra líder $\mathbf{u}$	Síndrome $S(\mathbf{u})$
00000	000
10000	011
01000	111
00100	100
00010	010
00001	001
* 00110	110
* 01010	101

y $$H^T = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(i) $\mathbf{w}_1 = 01011$:

$$S(\mathbf{w}_1) = \mathbf{w}_1 H^T = (01011) \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 100 = S(00100)$$

Decodificamos por $01011 - 00100 = 01111$, tal y como hicimos en el Ejemplo 2.7.5.

(ii) $\mathbf{w}_2 = 10110$:

$$S(\mathbf{w}_2) = \mathbf{w}_2 H^T = (10110) \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = 101 = S(01010)$$

Decodificamos por $10110 - 01010 = 11100$, tal y como hicimos en el ejemplo 2.7.5.

Bibliografía del capítulo

Cory-Pless [2], Martínez [7], Ling-Xing [6], Simón [8].

Capítulo 3

Códigos cíclicos

3.1. Preliminares: polinomios sobre cuerpos finitos

Denotando con $\mathbb{F}_q^*$ a $\mathbb{F}_q \setminus \{0\}$, sabemos que $(\mathbb{F}_q^*, \cdot)$ es un grupo, pero podemos decir más:

Teorema 3.1.1. $(\mathbb{F}_q^*, \cdot)$ *es un grupo cíclico.*

La demostración del anterior teorema puede encontrarse en [2]. Llamamos *raíz primitiva* de $\mathbb{F}_q$ a cualquier generador γ de $\mathbb{F}_q^*$. Se deduce de la definición que

$$\mathbb{F}_q = \left\{0, 1 = \gamma^0, \gamma, \gamma^2, \ldots, \gamma^{q-2}\right\}$$

y que $\gamma^i = 1 \Leftrightarrow (q-1)|i$.

Diremos que $\xi \in \mathbb{F}_q$ es una *raíz n-ésima de la unidad* si $\xi^n = 1$.

Diremos que $\xi \in \mathbb{F}_q$ es una *raíz n-ésima primitiva de la unidad* si $\xi^n = 1$ y $\xi^s \neq 1$ para cada $0 < s < n$.

Por tanto, una raíz primitiva $\gamma \in \mathbb{F}_q$ es una raíz $(q-1)$-ésima primitiva de la unidad y $\mathbb{F}_q$ contendrá una raíz n-ésima primitiva de la unidad si y sólo si $n|(q-1)$, en cuyo caso será $\gamma^{(q-1)/n}$ tal raíz.

Dados un anillo A y un ideal I de A, sabemos que la relación de congruencia módulo I

$$a \equiv b \ (\text{mód } I) \Leftrightarrow b - a \in I$$

es una relación de equivalencia, con lo que

$$a + I = b + I \Leftrightarrow b - a \in I$$

$$(0 + I = b + I \Leftrightarrow b \in I)$$

y el conjunto de las clases de equivalencia $\frac{A}{I} = \{a + I : a \in A\}$ tiene estructura de anillo con las operaciones

$$(a + I) + (b + I) = (a + b) + I \qquad (a + I)(b + I) = (ab) + I.$$

A este anillo le llamamos *anillo cociente*.
Sean $f, g \in \mathbb{F}_q[x]$ con $f \neq 0$ y $m = deg(f)$. Como $\mathbb{F}_q[x]$ es un dominio euclídeo con función euclídea el grado deg, sabemos que existen $q, r \in \mathbb{F}_q[x]$ únicos tales que $g = fq + r$ con $deg(r) < m$ o $r = 0$. Se deduce entonces que $q - r \in (f)$ y que $g + (f) = r + (f)$. Por tanto,

$$\frac{\mathbb{F}_q[x]}{(f)} = \left\{ \sum_{i=0}^{m-1} a_i x^i : a_0, \ldots, a_{m-1} \in \mathbb{F}_q \right\}$$

y consideraremos el representante canónico de un polinomio arbitrario $g \in \mathbb{F}_q[x]$ como el resto de dividir g entre f. Si además f es irreducible en $\mathbb{F}_q[x]$, entonces (f) es maximal y $\frac{\mathbb{F}_q[x]}{(f)}$ es un cuerpo con q^m elementos.

Ejemplo 3.1.2. Como $f = 1 + x + x^3$ es irreducible en $\mathbb{F}_2[x]$ por no tener raíces en $\mathbb{F}_2$, es

$$\mathbb{F}_8 \cong \mathbb{F}_2[x]/(f) = \left\{0, 1, x, 1+x, x^2, 1+x^2, x+x^2, 1+x+x^2\right\}.$$

Si llamamos α a una raíz de f, tenemos $1 + \alpha + \alpha^3 = 0$, luego

$$\alpha^3 = 1 + \alpha, \qquad \alpha^4 = \alpha + \alpha^2, \qquad \alpha^5 = 1 + \alpha + \alpha^2, \qquad \alpha^6 = 1 + \alpha^2$$

y, en este caso, cualquier raíz de f es una raíz primitiva de $\mathbb{F}_8$.

Dados $\mathbb{F}_q$ y una extensión $\mathbb{F}_{q^t}$, sabemos por Proposición 1.1.3 que cada $\alpha \in \mathbb{F}_{q^t}$ es una raíz de $x^{q^t} - x$. Podemos considerar entonces el polinomio mónico $Irr(\alpha; \mathbb{F}_q) \in \mathbb{F}_q[x]$ de grado mínimo que se anula en α. Llamaremos *polinomio irreducible de α sobre $\mathbb{F}_q$* a tal polinomio.

Demostraremos la siguiente proposición dado que contiene propiedades interesantes sobre los polinomios irreducibles, además de no ser complicada.

Proposición 3.1.3. *Sea $\mathbb{F}_{q^t}$ una extensión de $\mathbb{F}_q$, $\alpha \in \mathbb{F}_{q^t}$ y $n = deg(Irr(\alpha; \mathbb{F}_q))$. Entonces:*

(i) $Irr(\alpha; \mathbb{F}_q)$ es irreducible sobre $\mathbb{F}_q$.

(ii) Si $g \in \mathbb{F}_q[x]$ cumple $g(\alpha) = 0$, entonces $Irr(\alpha; \mathbb{F}_q) | g$.

(iii) $Irr(\alpha; \mathbb{F}_q)$ es único.

(iv) $\{1, \alpha, \ldots, \alpha^{n-1}\}$ es una base de $\mathbb{F}_q(\alpha)$ sobre $\mathbb{F}_q$.

(v) $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = n$.

Demostración. $\boxed{(i)}$ Si fuera reducible no sería de grado mínimo.

$\boxed{(ii)}$ El conjunto $\{f \in \mathbb{F}_q[x] : f(\alpha) = 0\}$ es un ideal de $\mathbb{F}_q[x]$, que es un dominio de ideales principales; debe ser $Irr(\alpha; \mathbb{F}_q)$ un generador de tal ideal para no tener una contradicción con (i). Entonces

$$g(\alpha) = 0 \Rightarrow g \in (Irr(\alpha; \mathbb{F}_q)) \Rightarrow Irr(\alpha; \mathbb{F}_q) | g.$$

$\boxed{(iii)}$ Si hubiera otro h mónico y de grado mínimo, se tendría que $Irr(\alpha;\mathbb{F}_q)-h$, de menor grado que $Irr(\alpha;\mathbb{F}_q)$, se anula en α, luego es divisible por $Irr(\alpha;\mathbb{F}_q)$ y, en consecuencia, son iguales.

$\boxed{(iv)}$

$$\sum_{i=0}^{n-1} a_i\alpha^i = 0 \Rightarrow g(\alpha) = 0 \text{ con } g = \sum_{i=0}^{n-1} a_i x^i \Rightarrow Irr(\alpha;\mathbb{F}_q)|g,$$

luego $g = 0$ por ser $deg(g) < n$ y $1, \alpha, \ldots, \alpha^{n-1}$ son linealmente independientes.

Sea ahora $\beta \in \mathbb{F}_q(\alpha)$. Entonces existe $h \in \mathbb{F}_q[x]$ con $h(\alpha) = \beta$. Por el algoritmo d la división existen $q, r \in \mathbb{F}_q[x]$ tales que $h = fq + r$ con $deg(r) < n$ o $r = 0$. Entonces

$$\beta = h(\alpha) = f(\alpha)q(\alpha) + r(\alpha) = r(\alpha) \in \left\langle 1, \alpha, \ldots, \alpha^{n-1} \right\rangle,$$

luego $\{1, \alpha, \ldots, \alpha^{n-1}\}$ es un conjunto de generadores.

$\boxed{(v)}$ Es consecuencia de (iv). □

Si dado $f \in \mathbb{F}_q[x]$ irreducible y de grado m llamamos α a una raíz de f en una extensión de $\mathbb{F}_q$, tenemos por la Proposición 3.1.3 (iv) que

$$\mathbb{F}_q(\alpha) = \left\{ \sum_{i=0}^{m-1} a_i\alpha^i : a_0, \ldots, a_{m-1} \in \mathbb{F}_q \right\} \cong \mathbb{F}_{q^m},$$

lo cual nos permite representar a los elementos de $\mathbb{F}_{q^m}$ de dos formas: bien como polinomios en α, bien como potencias de una raíz primitiva γ. Con esta correspondencia:

- Sumar potencias de α es más fácil si los representamos como polinomios.
- Multiplicar polinomios es más fácil si los representamos como potencias de γ.

Ejemplo 3.1.4. El polinomio $f = 1 + x + x^2 \in \mathbb{F}_5[x]$ es irreducible por no tener raíces en $\mathbb{F}_5$, luego $\mathbb{F}_{25} \cong \frac{\mathbb{F}_5[x]}{(f)}$. Sea α una raíz de f. Entonces $1 + \alpha + \alpha^2 = 0$ y

$$\alpha^2 = 4\alpha + 4 \qquad \alpha^3 = 4(4\alpha + 4) + 4\alpha = 16\alpha + 16 + 4\alpha = 1.$$

Ninguna raíz de f es raíz primitiva de $\mathbb{F}_{25}$.

¿Por qué en Ejemplo 3.1.2 todas las raíces eran raíces primitivas y ahora no? En el Ejemplo 3.1.2 tenemos $\mathbb{F}_2(\alpha) \cong \mathbb{F}_8$, luego las raíces de f "están" en $\mathbb{F}_8^*$. Como $\mathbb{F}_8^*$ tiene 7 elementos, todos éstos tienen orden 7, luego son raíces primitivas de $\mathbb{F}_8$. Sin embargo, en el ejemplo anterior tenemos $\mathbb{F}_5(\alpha) \cong \mathbb{F}_{25}$ y los elementos de $\mathbb{F}_{25}^*$ no tienen por qué tener orden 24.

3.2. Definiciones

Definición 3.2.1. *Diremos que un subconjunto S de $\mathbb{F}_q^n$ es* cíclico *si se da la siguiente condición:*

$$(a_0, a_1, \ldots, a_{n-2}, a_{n-1}) \in S \;\Rightarrow\; (a_{n-1}, a_0, a_1, \ldots, a_{n-2}) \in S$$

Diremos que un código lineal $C \subseteq \mathbb{F}_q^n$ es cíclico *si C es cíclico somo subconjunto de $\mathbb{F}_q^n$*

Decimos que la palabra $(u_{n-r}, \ldots, u_{n-1}, u_0, u_1, \ldots, u_{n-r-1})$ se obtiene de la palabra $(u_0, \ldots, u_{n-1}) \in \mathbb{F}_q^n$ desplazando cíclicamente r posiciones.

Proposición 3.2.2. *El código dual de un código cíclico también es cíclico.*

Demostración. Sea $(v_0, \ldots, v_{n-1}) \in C^\perp$. Entonces $(v_0, \ldots, v_{n-1})\cdot\mathbf{c} = 0$ para todo $\mathbf{c} \in C$. Sea $\mathbf{c} = (c_0, \ldots, c_{n-1})$ una palabra arbitraria de C. Tenemos

$$(v_{n-1}, v_0, \ldots, v_{n-2})(c_0, \ldots, c_{n-1}) = (v_0, \ldots, v_{n-1})(c_1, \ldots, c_{n-1}, c_0) = 0$$

Así que $(v_{n-1}, v_0, \ldots, v_{n-2}) \in C^\perp$. □

Ejemplos 3.2.3. Son códigos cíclicos:

(i) $\{\mathbf{0}\}$, $\{(\lambda, \ldots, \lambda) : \lambda \in \mathbb{F}_q\}$, $\mathbb{F}_q^n$.

(ii) El $[3, 2, 2]$-código lineal $\{000, 110, 101, 011\}$ sobre $\mathbb{F}_2$.

Como hemos comentado, el representante canónico de un polinomio arbitrario $f \in \mathbb{F}_q[x]$ en el anillo cociente

$$\mathbb{F}_{q,n}[x] = \frac{\mathbb{F}_q[x]}{(x^n - 1)} = \left\{ \sum_{i=0}^{n-1} a_i x^i : a_0, \ldots, a_{n-1} \in \mathbb{F}_q \right\}$$

es el resto de dividir f entre $x^n - 1$ y las operaciones en este anillo son las usuales del anillo cociente. Nótese que en este anillo se tiene $x^n = 1$ siguiendo la notación de la sección anterior. De ahora en adelante y para cualquier polinomio $g \in \mathbb{F}_q[x]$, denotaremos por (g) al ideal generado por g en $\mathbb{F}_q[x]$ y por $\langle g \rangle$ al ideal generado por g en $\mathbb{F}_{q,n}[x]$.

Sabemos que $\mathbb{F}_q^n$ y $\mathbb{F}_{q,n}[x]$ son $\mathbb{F}_q$-espacios vectoriales isomorfos bajo la correspondencia de bases

$$\begin{array}{rcl} \pi : \mathbb{F}_q^n & \to & \mathbb{F}_{q,n}[x] \\ (1, 0, \ldots, 0) & \rightsquigarrow & 1 \\ (0, 1, \ldots, 0) & \rightsquigarrow & x \\ & \vdots & \\ (0, 0, \ldots, 1) & \rightsquigarrow & x^{n-1}. \end{array}$$

Así, podemos identificar las palabras con los polinomios. Nótese que, lo que en una palabra es hacer un desplazamiento cíclico de una posición (también llamado *shift*), en un polinomio es "multiplicar por x":

$$(a_0, a_1, \ldots, a_{n-1}) \overset{\pi}{\rightsquigarrow} a_0 + a_1 x + \ldots + a_{n-1} x^{n-1} \overset{\cdot x}{\rightsquigarrow} a_{n-1} + a_0 x + \ldots + a_{n-2} x^{n-1} \overset{\pi^{-1}}{\rightsquigarrow} (a_{n-1}, a_0, \ldots, a_{n-2})$$

$a_0+a_1x+\ldots+a_{n-1}x^{n-1} \overset{\pi^{-1}}{\rightsquigarrow} (a_0, a_1, \ldots, a_{n-1}) \overset{shift}{\rightsquigarrow} (a_{n-1}, a_0, \ldots, a_{n-2}) \overset{\pi}{\rightsquigarrow} a_{n-1}+a_0x+\ldots+a_{n-2}x^{n-1}$

Este hecho es clave en el teorema que veremos a continuación.

3.3. Polinomios generadores

Teorema 3.3.1. *Para cualquier subconjunto no vacío $C \subseteq \mathbb{F}_q^n$ se tiene que C es un código cíclico si y sólo si $\pi(C)$ es un ideal de $\mathbb{F}_{q,n}[x]$.*

Demostración. $\boxed{\Rightarrow}$ Si $a = \sum_{i=0}^{n-1} a_i x^i$, $b = \sum_{i=0}^{n-1} b_i x^i \in \pi(C)$, entonces $\mathbf{a} = (a_0, \ldots, a_{n-1})$ $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in C$, luego $\mathbf{a} \pm \mathbf{b} \in C$ y $a \pm b \in \pi(C)$.

Comentamos que "multiplicar por x" equivale a "hacer un *shift*". Por tanto, dado $a \in \pi(C)$ arbitrario tenemos que $x^i a \in \pi(C)$ para cada $i = 0, \ldots, n-1$. Sea ahora $r = \sum_{i=0}^{n-1} r_i x^i \in \mathbb{F}_{q,n}[x]$. Entonces $ra = \sum_{i=0}^{n-1} r_i x^i a$. Como $r_i \in \mathbb{F}_q, x^i a \in \pi(C)$ y $\pi(C)$ es un $\mathbb{F}_q$-espacio vectorial, se tiene $r_i x^i a \in \pi(C)$ para cada $i = 0, \ldots, n-1$, luego $ra \in \pi(C)$.

$\boxed{\Leftarrow}$ Sean $\alpha, \beta \in \mathbb{F}_q$ y $\mathbf{a}, \mathbf{b} \in C$. Por hipótesis, $\alpha\pi(\mathbf{a}) + \beta\pi(\mathbf{b}) \in \pi(C)$. Ahora bien, $\alpha\pi(\mathbf{a}) + \beta\pi(\mathbf{b}) = \pi(\alpha\mathbf{a} + \beta\mathbf{b})$, luego $\alpha\mathbf{a} + \beta\mathbf{b} \in C$ y C es un código lineal.

La propiedad de ser cíclico se obtiene del hecho de que "hacer un *shift*" equivale a "multiplicar por x". □

Ejemplos 3.3.2.

(i) $C = \{000, 111, 222\} \subseteq \mathbb{F}_3^3$ es un código cíclico, luego $\pi(C) = \{0, 1+x+x^2, 2+2x+2x^2\}$ es un ideal de $\mathbb{F}_{3,3}[x]$.

(ii) $I = \{0, 1+x^2, x+x^3, 1+x+x^2+x^3\}$ es un ideal de $\mathbb{F}_{2,4}[x]$, luego $\pi^{-1}(I) = \{0000, 1010, 0101, 1111\} \subseteq \mathbb{F}_2^4$.

(iii) Los códigos cíclicos $\{\mathbf{0}\}$ y $\mathbb{F}_q^n$ se corresponden con los ideales $\{0\}$ y $\mathbb{F}_{q,n}[x]$, respectivamente.

Observación 3.3.3. Que hayamos relacionado los códigos cíclicos con los idales de $\mathbb{F}_{q,n}[x]$ no quiere decir que un ideal arbitrario de $\mathbb{F}_{q,n}[x]$ esté asociado a un sólo código. Concretamente, cada ideal de $\mathbb{F}_{q,n}[x]$ está asociado a una familia de códigos: el correspondiente cíclico que enuncia el teorema anterior y todos sus códigos equivalentes. Así, si σ es una permutación en las posiciones de un código cíclico, la composición $\pi \circ \sigma^{-1}$ asocia un código no necesariamente cíclico con un ideal de $\mathbb{F}_{q,n}[x]$. Ejemplo de esto es la permutación $\sigma = (1\ 4)$ y el código $C' = \{\mathbf{0}, 0011, 1100, 1111\}$. Se tiene $C = (C')^\sigma = \{\mathbf{0}, 1010, 0101, 1111\}$ y el ideal I de (ii) del ejemplo anterior.

Sea la sucesión exacta corta $0 \to (x^n - 1) \to \mathbb{F}_q[x] \overset{\eta}{\to} \mathbb{F}_{q,n}[x] \to 0$ y sea I un ideal no nulo de $\mathbb{F}_{q,n}[x]$. Por el teorema de la correspondencia, $\eta^{-1}(I)$ es un ideal de $\mathbb{F}_q[x]$ que contiene a $(x^n - 1)$, y como $\mathbb{F}_q[x]$ es un dominio euclídeo con función euclídea el grado *deg*, existirá $g \in \mathbb{F}_q[x]$ tal que $(g) = \eta^{-1}(I)$ y podemos suponerlo mónico de grado mínimo. Evidentemente $g | x^n - 1$ y además g es único, pues g' también mónico de grado mínimo

implica que $g - g'$ por un escalar adecuado es un polinomio mónico de menor grado que g en (g), lo cual es imposible. Como $deg(g) < n$, el resto de dividir g entre $x^n - 1$ es el propio g, así que identificamos $\eta(g) = g + (x^n - 1) = g$ siguiendo nuestra notación y tenemos $\langle g \rangle = I$. Lo reenunciamos como un corolario.

Corolario 3.3.4. *Para cada ideal no nulo I de $\mathbb{F}_{q,n}[x]$ existe un único polinomio $g \in I$ mónico, de grado mínimo y cumpliendo $\langle g \rangle = I$ y $g|x^n - 1$.*

Al polinomio g del corolario anterior se le llama el *polinomio generador* de I.

Dado un código cíclico C, al polinomio generador de $\pi(C)$ se le llama también el *polinomio generador* de C.

Ejemplos 3.3.5. En los Ejemplos 3.3.2 tenemos que

(i) $1 + x + x^2$ es el polinomio generador de C y además $x^3 - 1 = (x^2 + x + 1)(x - 1)$.

(ii) $1 + x^2$ es el polinomio generador de I y además $x^4 - 1 = (x^2 + 1)(x^2 - 1)$.

(iii) 1 es el polinomio generador de $\mathbb{F}_q^n$.

Se sigue así que hay una correspondencia biyectiva

$$\begin{array}{rcl} \{\text{códigos cíclicos en } \mathbb{F}_q^n\} & \leftrightarrow & \{\text{divisores mónicos de } x^n - 1 \in \mathbb{F}_q[x]\} \\ C & \rightsquigarrow & \text{el polinomio generador } g \text{ de } C \\ \pi^{-1}(\langle g \rangle) & \leftsquigarrow & g \end{array}$$

en la que se tiene

$$\begin{array}{rcl} \mathbb{F}_q^n & \leftrightsquigarrow & 1 \\ \{\mathbf{0}\} & \leftrightsquigarrow & x^n - 1 \ (\equiv 0 \ (\text{mód } x^n - 1)) \end{array}$$

Ejemplo 3.3.6. Encontrar todos los códigos cíclicos en $\mathbb{F}_2^6$.

Factorizamos $x^6 - 1 \in \mathbb{F}_2[x]$:

$$x^6 - 1 = (x^3 - 1)(x^3 + 1) = (x^3 + 1)^2 = (x + 1)^2(x^2 + x + 1)^2$$

De la biyección anterior observamos que los códigos cíclicos en $\mathbb{F}_2^6$ son los generados por los divisores mónicos de $x^6 - 1$, que son a su vez los divisores mónicos de $(1+x)^2(1+x+x^2)^2$:

$$\begin{array}{ccc} 1 & (1 + x + x^2) & (1 + x + x^2)^2 \\ 1 + x & (1 + x)(1 + x + x^2) & (1 + x)(1 + x + x^2)^2 \\ (1 + x)^2 & (1 + x)^2(1 + x + x^2) & (1 + x)^2(1 + x + x^2)^2 \end{array}$$

Veamos algunos de los códigos asociados a los anteriores polinomios:

Si tomamos la palabra 101010 (equivalentemente, el polinomio $(1+x+x^2)^2 = 1+x^2+x^4$) y desplazamos cíclicamente una posición (equivalentemente, hacemos el producto

$x(1+x^2+x^4))$, obtenemos 010101 $(x+x^3+x^5)$. Para que el código sea lineal, deben estar $\mathbf{0}$ y todas las sumas realizadas con las palabras

$$\mathbf{0}, \qquad 101010 \quad \text{y} \quad 010101,$$

esto es, la palabra 111111 $(1+x+x^2+x^3+x^4+x^5)$. Tenemos por tanto uno de los nueve códigos cíclicos de $\mathbb{F}_2^6$: $\{\mathbf{0}, 101010, 010101, 111111\}$.

Como comentamos en la anterior biyección, otros dos códigos cíclicos son $\mathbb{F}_2^6$ (generado por 1) y $\{\mathbf{0}\}$ (generado por $(1+x)^2(1+x+x^2)^2 = x^6-1$).

Otro código cíclico sencillo es $\{\mathbf{0}, 111111\}$. Sabiendo que $1+x+x^2+x^3+x^4+x^5$ debe ser divisible por $x+1$, obtenemos de las igualdades

$$(1+x+x^2+x^3+x^4+x^5)(1+x) = x^6-1 = (1+x)^2(1+x+x^2)^2$$

que es el código generado por $(1+x)(1+x+x^2)^2$.

Observación 3.3.7. Si

$$x^n - 1 = \prod_{i=1}^{r} p_i^{e_i}$$

es la factorización de $x^n-1 \in \mathbb{F}_q[x]$ en polinomios $p_1, \ldots, p_r \in \mathbb{F}_q[x]$ distintos, mónicos e irreducibles con $e_1, \ldots, e_r \in \mathbb{Z}^+$, entonces hay $\prod_{i=1}^{r}(e_i+1)$ códigos cíclicos en $\mathbb{F}_q^n$.

Podemos dar más información de C a partir de su polinomio generador.

Teorema 3.3.8. *Sea C un código cíclico en $\mathbb{F}_q^n$ y g su polinomio generador. Si $deg(g) = n-k$, entonces $dim(C) = k$.*

Demostración. Sea $A = \{gc : c \in \mathbb{F}_{q,n}[x], deg(c) \leq k-1\}$. De la definición se tiene $A \subseteq \langle g \rangle$. Para ver la inclusión contraria, sea ga con $a \in \mathbb{F}_{q,n}[x]$ un elemento arbitrario de $\langle g \rangle$. Por el algoritmo de la división,

$$ga = u(x^n-1) + v$$

para ciertos $u, v \in \mathbb{F}_q[x]$ con $deg(v) < n$, luego

$$ga - u(x^n-1) = v$$

y $g|v$. Escribiendo $v = gb$ con $b \in \mathbb{F}_q[x]$, resulta que $deg(b) < k$, luego $b \in \mathbb{F}_{q,n}[x]$ y $v \in A$. Así que $A \supseteq \langle g \rangle$ y acabamos de probar que $A = \langle g \rangle$. Veamos que $\{g, xg, \ldots, x^{k-1}g\}$ es una base:
-Linealmente independientes: Inmediato del hecho de que $deg(g) = n-k$.
-Conjunto generador: Sea $a \in \langle g \rangle = A$. Existe $c = \sum_{i=0}^{k-1} c_i x^i$ tal que

$$a = gc = \sum_{i=0}^{k-1} c_i x^i g \in \langle g, xg, \ldots, x^{k-1}g \rangle$$

□

El anterior teorema nos da una condición necesaria y suficiente para hallar códigos cíclicos en $\mathbb{F}_q^n$ de una determinada dimensión.

Ejemplos 3.3.9.

(i) Hallar los $[7,3]$-códigos cíclicos sobre $\mathbb{F}_2$.

Sabemos que los divisores mónicos de x^7-1 (considerado como elemento de $\mathbb{F}_2[x]$) son polinomios generadores de códigos cíclicos en $\mathbb{F}_2^7$. Factorizamos x^7-1:

$$x^7-1=(1+x)(1+x+x^2+x^3+x^4+x^5+x^6)=(1+x)(1+x+x^3)(1+x^2+x^3)$$

Hay entonces dos divisores mónicos de grado $7-3=4$:

$$g_1=(1+x)(1+x+x^3)=1+x^2+x^3+x^4$$
$$g_2=(1+x)(1+x+x^3)=1+x+x^2+x^4$$

Cada polinomio generará un $[7,3]$-código cíclico sobre $\mathbb{F}_2$ y no habrá más códigos de estas características, puesto que el resto de divisores mónicos de x^7-1 tienen otros grados distintos de 4 y, en consecuencia, generarán códigos cíclicos en $\mathbb{F}_2^7$ de otras dimensiones.

(ii) Hallar los $[7,2]$-códigos cíclicos sobre $\mathbb{F}_3$.

Sabemos que los divisores mónicos de x^7-1 (considerado como elemento de $\mathbb{F}_3[x]$) son polinomios generadores de códigos cíclicos en $\mathbb{F}_3^7$. Factorizamos x^7-1:

$$x^7-1=(2+x)(1+x+x^2+x^3+x^4+x^5+x^6)$$

Como no hay ningún divisor mónico de x^7-1 de grado $7-2=5$, afirmamos que no existen $[7,2]$-códigos cíclicos sobre $\mathbb{F}_3$.

3.4. Matrices generadoras y matrices de control de paridad

A partir del polinomio generador de un código cíclico se puede construir una matriz generadora de dicho código.

Teorema 3.4.1. *Sea $g\in\mathbb{F}_{q,n}[x]$ de grado $n-k$ y con coeficientes $g_0,g_1,\ldots,g_{n-k}$ el polinomio generador de un código cíclico en $\mathbb{F}_q^n$. Entonces*

$$G=\begin{pmatrix} — & x^0g & — \\ & \vdots & \\ & \vdots & \\ — & x^{k-1}g & — \end{pmatrix}=\begin{pmatrix} g_0 & \ldots & \ldots & g_{n-k} & 0 & \ldots & 0 \\ 0 & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & 0 \\ 0 & \ldots & 0 & g_0 & \ldots & \ldots & g_{n-k} \end{pmatrix}_{k\times n}$$

es una matriz generadora de dicho código.

Demostración. Es consecuencia del Teorema 3.3.8. □

Ejemplo 3.4.2. Consideremos el $[7, 4]$-código cíclico C sobre $\mathbb{F}_2$ con polinomio generador $g = 1 + x^2 + x^3$. Por el teorema anterior, una matriz generadora de C es

$$G = \begin{pmatrix} — \ x^0 g \ — \\ \vdots \\ \vdots \\ — \ x^3 g \ — \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Sustituyendo la segunda fila por la suma de ésta con la cuarta y sustituyendo la primera fila por la fila de ésta con la tercera y la cuarta, conseguimos una matriz generadora en forma estándar:

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

Observación 3.4.3. El que hayamos obtenido una matriz generadora en forma estándar en el anterior ejemplo no ha sido suerte: como $g_0 \neq 0$ por dividir g a $x^n - 1$, podemos tomar los g_0 del Teorema 3.4.1 como pivotes y operar hasta conseguir una matriz generadora en forma estándar.

Dado el polinomio generador g de un código cíclico C, si quisiéramos calcular una matriz de control de paridad de C tendríamos que:

- o bien calcular una matriz generadora y, a partir de ésta, una matriz de control de paridad;
- o bien calcular el código dual $C^\perp$ y, a partir de éste o de su polinomio generador, calcular una matriz generadora (la cual sería, por definición, una matriz de control de paridad de C).

Vista la simplicidad que nos proporcionan los polinomios generadores, nos preguntamos si, a partir de g, podemos hallar el polinomio generador del código dual $C^\perp$.

Definición 3.4.4. *Sea $h = a_0 + a_1 x + \ldots + a_k x^k \in \mathbb{F}_q[x]$ de grado k. Definimos el* polinomio recíproco *de h como*

$$h_R = x^k h\left(\frac{1}{x}\right) = a_k + a_{k-1} x + \ldots + a_0 x^k \in \mathbb{F}_q[x]$$

Observación 3.4.5. Si $h \in \mathbb{F}_q[x]$ es divisor de $x^n - 1 \in \mathbb{F}_q[x]$, entonces también lo es h_R.

Ejemplo 3.4.6. El polinomio recíproco de $h = 1 + 2x + 3x^5 + x^7 \in \mathbb{F}_5[x]$ es $h_R = 1 + 3x^2 + 2x^6 + x^7$.

Teorema 3.4.7. *Sea $g \in \mathbb{F}_{q,n}[x]$ el polinomio generador de un $[n, k]$-código cíclico C sobre $\mathbb{F}_q$ y sea*

$$h = \frac{x^n - 1}{g} = h_0 + h_1 x + \ldots + h_k x^k \in \mathbb{F}_q[x].$$

Entonces $h_0 \neq 0$ y $h_0^{-1} h_R$ es el polinomio generador de $C^\perp$.

Demostración. Denotando $g = \sum_{i=0}^{n-1} g_i x^i$ con $deg(g) \leq n-1$ y $h = \sum_{i=0}^{n-1} h_i x^i$ con $deg(h) = k \leq n-1$, tenemos $gh = x^n - 1$ (así que $g_0 h_0 = -1$ y $h_0 \neq 0$) y

$$\begin{array}{rcl} gh & \equiv & (g_0h_0 + g_1h_{n-1} + \ldots + g_{n-1}h_1) \\ & + & (g_0h_1 + g_1h_0 + \ldots + g_{n-1}h_2)x \\ & & \vdots \\ & + & (g_0h_{n-1} + g_1h_{n-2} + \ldots + g_{n-1}h_0)x^{n-1} \equiv 0 \ (\text{mód } x^n - 1), \end{array}$$

luego

$$\begin{array}{c} (g_0, g_1, \ldots, g_{n-1})(h_{n-1}, \ldots, h_0) = 0 \\ (g_{n-1}, g_0, \ldots, g_{n-2})(h_{n-1}, \ldots, h_0) = 0 \\ \vdots \\ (g_1, g_2, \ldots, g_{n-1}, g_0)(h_{n-1}, \ldots, h_0) = 0 \end{array} \tag{3.1}$$

Como las palabras que están en (3.1) multiplicando por la izquierda forman una base de C por el Teorema 3.4.1, $(h_{n-1}, \ldots, h_0) \in C^\perp$. Y como $C^\perp$ es cíclico, entonces también $(h_k, \ldots, h_0, h_{n-1}, \ldots, h_{k+1}) \in C^\perp$, es decir, $h_R \in C^\perp$. Como $dim(C^\perp) = n-k$ y $deg(h_R) = deg(h) = k$, tenemos que $\{h_R, xh_R, \ldots, x^{n-k-1}h_R\}$ es una base de $C^\perp$ y, por tanto, que $h_0^{-1}h_R$ es el polinomio generador de $C^\perp$. □

Definición 3.4.8. *Al polinomio $h_0^{-1}h_R$ del anterior teorema lo llamaremos el* polinomio de control de paridad *de C.*

Corolario 3.4.9. *Sea $g \in \mathbb{F}_{q,n}[x]$ el polinomio generador de un $[n, k]$-código cíclico C sobre $\mathbb{F}_q$ y sea*

$$h = \frac{x^n - 1}{g} = h_0 + h_1 x + \ldots + h_k x^k \in \mathbb{F}_q[x].$$

Entonces

$$H = \begin{pmatrix} — \ x^0 h_R \ — \\ \vdots \\ \vdots \\ — \ x^{n-k-1} h_R \ — \end{pmatrix} = \begin{pmatrix} h_k & \ldots & \ldots & h_0 & 0 & \ldots & 0 \\ 0 & \ddots & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & 0 \\ 0 & \ldots & 0 & h_k & \ldots & \ldots & h_0 \end{pmatrix}_{(n-k)\times n}$$

es una matriz de control de paridad de C.

Como $dim(C) = k$, entonces $deg(g) = n - k$ y $deg(h) = k$, por lo que $h_k \neq 0$ y, pivotando en esos coeficientes h_k, deducimos que todo código cíclico posee una matriz de control en la forma $(I_{n-k}|A)$.

Ejemplo 3.4.10. Consideremos de nuevo el $[7, 4]$-código cíclico C sobre $\mathbb{F}_2$ con polinomio generador $g = 1 + x^2 + x^3$.

$$\begin{array}{rrrrrrrrl}
x^7 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & \underline{\lfloor x^3 + x^2 + 1} \\
0 & +x^6 & 0 & +x^4 & & & & -1 & x^4 + x^3 + x^2 + 1 \\
 & 0 & +x^5 & +x^4 & +x^3 & & & -1 & \\
 & & 0 & 0 & +x^3 & +x^2 & & -1 & \\
 & & & & & & & \underline{0} &
\end{array}$$

Sea $h = \dfrac{x^7 - 1}{g} = 1 + x^2 + x^3 + x^4$. Entonces $h_R = 1 + x + x^2 + x^4$ es el polinomio de control de paridad de C y

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

es una matriz de control de paridad de C.

3.5. Decodificación de códigos cíclicos

Todos los síndromes que calculemos en esta sección serán respecto de una matriz de control de paridad en la forma $H = (I_{n-k}|A)$. Denotaremos por $(w(\text{mód } g))$ o por $\overline{w}$ al resto de dividir w por un polinomio generador dado g.

Teorema 3.5.1. *Sea $H = (I_{n-k}|A)$ una matriz de control de paridad de un código cíclico C sobre $\mathbb{F}_q$ con polinomio generador g. Entonces*

$$\forall \mathbf{w} \in \mathbb{F}_q^n, \quad (S(\mathbf{w}) =)\ \mathbf{w}H^T = (w(\text{mód } g)),$$

donde con el signo '=' estamos connotando también la identificación "palabra — polinomio".

Demostración. Sabemos por hipótesis que H es matriz de control, luego $G = (-A^T|I_k)$ es matriz generadora. Identificando G con la matriz

$$\left(\begin{array}{ccc|ccc} — & -a_0 & — & x^{n-k} & & \\ & \vdots & & & \ddots & \\ — & -a_{k-1} & — & & & x^n \end{array}\right) \qquad (a_i \in \mathbb{F}_q[x]),$$

tenemos que $x^{n-k+i} - a_i \in C$ para cada $i = 0, \ldots, k-1$ y así, $a_i = x^{n-k+i} - q_i g$ para cierto $q_i \in \mathbb{F}_{q,n}[x]$. Sea $w = \sum_{i=0}^{n-1} w_i x^i$. Identificando H con

$$\left(\begin{array}{ccc|ccc} 1 & & & | & & | \\ & x & & | & & | \\ & & \ddots & a_0 & \ldots & a_{k-1} \\ & & & | & & | \\ & & x^{n-k-1} & | & & | \end{array}\right)$$

tenemos

$$\begin{aligned} S(\mathbf{w}) &= \mathbf{w}H^T = \sum_{i=0}^{n-k-1} w_i x^i + \sum_{j=0}^{k-1} w_{n-k+j} a_j \\ &= \sum_{i=0}^{n-k-1} w_i x^i + \sum_{j=0}^{k-1} w_{n-k+j}(x^{n-k+j} - q_j g) \\ &= w - \left(\sum_{j=0}^{k-1} w_{n-k+j} q_j\right) g \equiv \overline{w} \ (\text{mód } g) \end{aligned}$$

Como $S(\mathbf{w})$ tiene grado menor que $deg(g) = n-k$ y $g | S(\mathbf{w}) - \overline{w}$, se tiene que $S(\mathbf{w}) - \overline{w} = 0$ y que $S(\mathbf{w}) = \overline{w}$. □

Ejemplo 3.5.2. Consideremos de nuevo el $[7, 4]$-código cíclico C sobre $\mathbb{F}_2$ con polinomio generador $g = 1 + x^2 + x^3$ y matriz de control de paridad

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

Sustituyendo la primera fila por la suma de ésta con la segunda y la segunda fila por la suma de ésta con la tercera, obtenemos

$$H = (I_3|A) = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Sea $\mathbf{w} = 0110110$. Por un lado,

$$S(\mathbf{w}) := \mathbf{w}H^T = (0110110) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = 010 \ = \ x$$

Por otro,

$$\begin{array}{cccccc|l} x^5 & x^4 & 0 & x^2 & x & 0 & \underline{x^3+x^2+1} \\ 0 & 0 & 0 & 0 & \underline{x} & 0 & x^2 \end{array},$$

luego $(w(\text{mód } g)) = x = 010$ y, tal y como afirma el teorema, coinciden. Aprovechamos para calcular $d(C)$: las columnas 2, 4 y 5 de H suman $\mathbf{0}^T$, luego son linealmente dependientes. Nos preguntamos si todo par de columnas de H es linealmente independiente. Como el cuerpo sobre el que está C es $\mathbb{F}_2$, esto es equivalente a preguntarse si la suma de cualesquiera dos columnas de H es distinta de $\mathbf{0}^T$, es decir, si no hay dos columnas iguales. La respuesta es afirmativa, luego $d(C) = 3$.

Del anterior teorema se deduce que el resto $s = (w(\text{mód } g))$ y w tienen el mismo síndrome, luego están en la misma clase lateral y, en consecuencia, $\mathbf{w} - \mathbf{s}$ es una palabra del código. Se tiene el siguiente corolario.

Corolario 3.5.3. *Sean g el polinomio generador de un código cíclico C, $\mathbf{w} \in \mathbb{F}_q^n$ una palabra recibida y $s = (w(\text{mód } g))$. Si $wt(\mathbf{s}) \leq \left\lfloor \frac{d(C)-1}{2} \right\rfloor$, entonces $\mathbf{s}$ es el patrón de error (la palabra líder) de $\mathbf{w} + C$, por lo que decodificamos $\mathbf{w}$ por $\mathbf{w} - \mathbf{s}$ siguiendo la regla de decodificación por el vecino más cercano.*

Ejemplo 3.5.4. En el anterior ejemplo teníamos el polinomio generador $g = 1 + x^2 + x^3$, recibimos $\mathbf{w} = 0110110$ y calculamos el resto $s = (w(\text{mód } g)) = x$. Como

$$wt(\mathbf{s}) = 1 \leq \left\lfloor \frac{3-1}{2} \right\rfloor = 1,$$

decodificamos por $\mathbf{w} - \mathbf{s} = 0010110$.

Si recibimos ahora $\mathbf{w}_1 = 1011100$, tenemos

$$\begin{array}{cccccc|l} w_1 = x^4 & x^3 & x^2 & 0 & 1 & & \underline{x^2+x+1} \\ s_1 = 0 & 0 & \underline{x^2} & \underline{+x} & \underline{1} & & x \end{array}$$

Como $wt(\mathbf{s}_1) = 3 \not\leq 1$, no podemos afirmar que $\mathbf{s}_1$ es el patrón de error (la palabra líder) de $\mathbf{w}_1 + C$, por lo que habría que buscar dicha palabra líder $\mathbf{u}_1$ y decodificar $\mathbf{w}_1$ por $\mathbf{w}_1 - \mathbf{u}_1$.

Por suerte y como cabía esperar, hay un algoritmo de decodificiación para códigos cíclicos y éste se basa en el siguiente Lema.

Lema 3.5.5. *Sea C un $[n,k]$-código cíclico sobre $\mathbb{F}_q$ con polinomio generador g y sea*

$$s = s_0 + s_1 x + \ldots + s_{n-k-1} x^{n-k-1}$$

el síndrome $S(w)$ de una palabra recibida $\mathbf{w}$. Entonces el síndrome de $\mathbf{w}$ desplazada cíclicamente una posición es

$$S(xw) = xs - s_{n-k-1} g = xS(w) - s_{n-k-1} g$$

Demostración. Tenemos $w = qg + s$ para cierto $q \in \mathbb{F}_{q,n}[x]$ y que

$$xw = xqg + xs = (xq + s_{n-k-1})g + xs - s_{n-k-1}g$$

Como $deg(xs - s_{n-k-1}g) < n - k = deg(g)$ al ser g mónico, se tiene que $xs - s_{n-k-1}g = (xw(\text{mód } g)) = S(xw)$. □

Ejemplo 3.5.6. En el ejemplo 3.5.2 vimos que el síndrome de $w = x + x^2 + x^4 + x^5$ es $x(= 0 + x + 0x^2)$. Por el anterior lema:

$$S(xw) = xS(w) - 0g = x{\cdot}x = x^2$$
$$S(x^2w) = xS(xw) - 1g = x{\cdot}x^2 - (1 + x^2 + x^3) = 1 + x^2$$

Del anterior lema se sigue que

$$S(xw) = \overline{S(xw)} = \overline{xS(w) - s_{n-k-1}g} = \overline{xS(w)} - \overline{s_{n-k-1}g} = \overline{xS(w)}$$
$$S(x^2w) = \overline{xS(xw)} = \overline{x}\ \overline{xS(w)} = \overline{x^2S(w)}$$
$$\vdots$$
$$S(x^iw) = \overline{x^iS(w)} \text{ para todo } i = 0, \ldots, n-1,$$

por lo que, por ejemplo, para saber el síndrome de cualquier palabra de peso uno, basta con saber $S(1)$ y operar. Esto se traduce en que la tabla de síndromes queda reducida a una n-ésima parte.

Palabra líder **u**
1000...000
0100...000
⋮
0000...001
1100...000
0110...000
⋮
0000...011
1010...000
0101...000
⋮
0000...101
⋮

$\rightarrow$

Palabra líder **u**
l_1=1000...000
l_2=1100...000
l_3=1010...000
⋮

Sea ahora **w** una palabra recibida y supongamos que ocurren $\left\lfloor \frac{d-1}{2} \right\rfloor$ errores o menos. Por lo que acabamos de ver, existen $i \in \{0, \ldots, n-1\}$ y $k \in \{1, \ldots, r\}$ tales que $S(w) = \overline{x^{n-i}S(l_k)}$. Entonces

$$S(x^iw) = \overline{x^iS(w)} = \overline{x^i}\ \overline{x^{n-i}S(l_k)} = \overline{S(l_k)} = S(l_k)$$

y corregiríamos x^iw por $x^iw - l_k$. Multiplicando por x^{n-i}, corregimos w por $w - x^{n-i}l_k$. Veamos el algoritmo.

Algoritmo de decodificación para códigos cíclicos

Sea C un $[n, k, d]$-código cíclico sobre $\mathbb{F}_q$ con polinomio generador g y sea $\mathbf{w}$ una palabra recibida con patrón de error $\mathbf{e}$ tal que $wt(\mathbf{e}) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$.

Paso 1: Para cada $i = 0, 1, \ldots, n-1$, calcular $S(x^i w) = \overline{x^i w}$.

Paso 2: Encontrar $i_0 \in \{0, \ldots, n-1\}$ tal que $S(x^{i_0} w) = S(l_k)$ para cierta palabra líder l_k de la nueva tabla de decodificación por síndrome.

Paso 3: Llamar $e = x^{n-i_0} l_k$ y decodificar $\mathbf{w}$ por $\mathbf{w} - \mathbf{e}$.

Ejemplo 3.5.7. Seguimos en el polinomio generador $g = 1 + x^2 + x^3$ y la palabra $w_1 = 1 + x^2 + x^3 + x^4$ del Ejemplo 3.5.4. Hemos visto que el código es 1-corrector, luego, aplicando el algoritmo, tenemos que ir calculando $S(x^i w_1)$ hasta que alguno coincida con $S(l_1) = S(1) = 1$. Vimos que $S(x^0 w_1) = 1 + x + x^2$. Tenemos:

$$S(xw_1) = xS(x^0 w_1) - 1g = x(1 + x + x^2) - (1 + x^2 + x^3) = 1 + x \neq 1 = S(l_1)$$

$$S(x^2 w_1) = xS(xw_1) - 0g = x(1 + x) = x + x^2 \neq 1 = S(l_1)$$

$$S(x^3 w_1) = xS(x^2 w_1) - 1g = x(x + x^2) - (1 + x^2 + x^3) = 1 = S(l_1)$$

Llamamos $e = x^{7-3} l_1 = x^4 l_1 = 0000100$ y decodificamos $\mathbf{w}_1$ por

$$\mathbf{w}_1 - \mathbf{e} = 1011100 - 0000100 = 1011000$$

Bibliografía del capítulo

Asensio-Caruncho-Martínez [1], Cory-Pless [2], del Río-Simón-del Valle [3], García [4], Ling-Xing [6], Simón [8].

Capítulo 4

Códigos BCH

4.1. Preliminares: clases ciclotómicas

Dada una extensión $\mathbb{F}_{q^t}$ de $\mathbb{F}_q$ y $\alpha \in \mathbb{F}_{q^t}$, de (ii) de la Proposición 3.1.3 deducimos que $Irr(\alpha;\mathbb{F}_q)|x^{q^t}-x$, y como las raíces de $x^{q^t}-x$ son distintas por ser su derivada -1, entonces también las raíces de $Irr(\alpha;\mathbb{F}_q)$ son distintas. Si $\sigma \in Gal(\mathbb{F}_{q^t}|\mathbb{F}_q)$, sabemos que $0=\sigma(Irr(\alpha;\mathbb{F}_q)(\alpha))=Irr(\alpha;\mathbb{F}_q)(\sigma(\alpha))$, por lo que los conjugados de α son los $\sigma(\alpha)$, es decir, los $\alpha, \alpha^q, \alpha^{q^2}, \ldots, \alpha^{q^{r-1}}$ con r satisfaciendo $\alpha^{q^r}=\alpha$. Fijamos γ una raíz primitiva de $\mathbb{F}_{q^t}$. Entonces $\alpha=\gamma^s$ para algún s y $\gamma^{sq^r}=\gamma^s$, luego $\gamma^{sq^r-s}=1$ y $(q^t-1)|sq^r-s$. Así que $sq^r \equiv s \pmod{q^t-1}$ y esta repetición módulo q^t-1 nos lleva a la definición de *clase q-ciclotómica de s módulo* q^t-1

$$C_s=\left\{s, sq, \ldots, sq^{r-1}\right\} \pmod{q^t-1},$$

donde r es el entero positivo más pequeño tal que $sq^r \equiv s \pmod{q^t-1}$. Más adelante extenderemos esta definición para los divisores de q^t-1.

Observaciones 4.1.1.

(i) Dos clases q-ciclotómicas son iguales o disjuntas, por lo que las clases q-ciclotómicas forman una partición de $\{0,1,2,\ldots,q^t-2\}$.

(ii) $|C_s| \leq t$ para cada s.

(iii) Si s es coprimo con q^t-1, entonces $|C_s|=t$.

(iv) Si q^t-1 es primo, entonces $|C_s|=t$ para cada s.

Veamos un par de ejemplos.

Ejemplos 4.1.2.

(i) Clases 2-ciclotómicas módulo 15:

$$C_0=\{0\} \qquad C_1=\{1,2,4,8\} \qquad C_3=\{3,6,12,9\}$$

$$C_5 = \{5, 10\} \qquad C_7 = \{7, 14, 13, 11\}$$

El conjunto $\{0, 1, 3, 5, 7\}$ es un conjunto completo de representantes de las clases 2-ciclotómicas módulo 15.

(*ii*) Clases 3-ciclotómicas módulo 26:

$$\begin{array}{cccc} C_0 = \{0\} & C_1 = \{1, 3, 9\} & C_2 = \{2, 6, 18\} & C_4 = \{4, 12, 10\} \\ C_5 = \{5, 15, 19\} & C_7 = \{7, 21, 11\} & C_8 = \{8, 24, 20\} & C_{13} = \{13\} \\ & C_{14} = \{14, 16, 22\} & C_{17} = \{17, 23, 25\} & \end{array}$$

El conjunto $\{0, 1, 2, 4, 5, 7, 8, 13, 14, 17\}$ es un conjunto completo de representantes de las clases 3-ciclotómicas módulo 26.

Tenemos así que los conjugados de $Irr(\gamma^s; \mathbb{F}_q)$ son las potencias γ^i con $i \in C_s$. Lo enunciamos como un teorema.

Teorema 4.1.3. *Si γ es una raíz primitiva de $\mathbb{F}_{q^t}$, entonces*

$$Irr(\gamma^s; \mathbb{F}_q) = \prod_{i \in C_s} (x - \gamma^i),$$

donde C_s es la única clase q-ciclotómica de s módulo $q^t - 1$

Demostración. Puede encontrarse en [2]. □

Tenemos así la descomposición del irreducible de cualquier $\alpha \in \mathbb{F}_{q^t}$ sobre $\mathbb{F}_q$.

Ejemplo 4.1.4. En el Ejemplo 3.1.2 vimos la correspondencia "polinomio—potencia de α" que hay en $\mathbb{F}_8$:

$$\begin{array}{rcl@{\qquad}rcl} 0 & = & 0 & 1+\alpha & = & \alpha^3 \\ 1 & = & \alpha^0 & \alpha+\alpha^2 & = & \alpha^4 \\ \alpha & = & \alpha & 1+\alpha+\alpha^2 & = & \alpha^5 \\ \alpha^2 & = & \alpha^2 & 1+\alpha^2 & = & \alpha^6 \end{array}$$

Hallamos las clases 2-ciclotómicas módulo 7:

$$C_0 = \{0\} \qquad C_1 = \{1, 2, 4\} \qquad C_3 = \{3, 5, 6\}$$

Ya podemos hallar los polinomios irreducibles de las potencias de α:

$$\begin{aligned}
Irr(0;\mathbb{F}_2) &= x-0=x\\
Irr(\alpha^0;\mathbb{F}_2) &= x-\alpha^0=x+1\\
Irr(\alpha;\mathbb{F}_2) &= (x-\alpha)(x-\alpha^2)(x-\alpha^4)\\
&= x^3+x^2(-\alpha^4-\alpha^2-\alpha)+x(\alpha^6+\alpha^5+\alpha^3)-\alpha^7\\
&= x^3+x^2(\alpha+\alpha^2+\alpha^2+\alpha)+x(1+\alpha^2+1+\alpha+\alpha^2+1+\alpha)+1=x^3+x+1\\
Irr(\alpha^3;\mathbb{F}_2) &= (x-\alpha^3)(x-\alpha^5)(x-\alpha^6)\\
&= x^3+x^2(-\alpha^6-\alpha^5-\alpha^3)+x(\alpha^{11}+\alpha^9+\alpha^8)-\alpha^{14}\\
&= x^3+x^2(1+\alpha^2+1+\alpha+\alpha^2+1+\alpha)+x(\alpha^4+\alpha^2+\alpha)+1\\
&= x^3+x^2+x(\alpha+\alpha^2+\alpha^2+\alpha)+1=x^3+x^2+1
\end{aligned}$$

Ahora queremos encontrar la factorización de x^n-1 sobre $\mathbb{F}_q$ en factores irreducibles distintos (debemos suponer, pues, que q y n son coprimos) y para ello necesitamos saber qué extensión de $\mathbb{F}_q$ contiene una raíz n-ésima primitiva de la unidad.

Definición 4.1.5. *Definimos el* orden de q módulo n *como el menor* $a\in\mathbb{Z}^+$ *tal que* $q^a\equiv 1$ *(mód* n*). Lo denotaremos como* $ord_n(q)$*.*

Así, si $t=ord_n(q)$, tenemos por definición que t es el menor entero positivo tal que $n|(q^t-1)$, luego, en vista al comentario que hicimos tras la definición de raíz n-ésima primitiva de la unidad (página 27), $\mathbb{F}_{q^t}$ es la extensión más pequeña de $\mathbb{F}_q$ que contiene una raíz n-ésima primitiva de la unidad $\alpha=\gamma^d$ con $d=(q^t-1)/n$ y γ una raíz primitiva de $\mathbb{F}_{q^t}$. Y como $(\alpha^i)^n=(\alpha^n)^i=1$ $(i=0,\dots,n-1)$, resulta que $\mathbb{F}_{q^t}$ es el cuerpo de descomposición de x^n-1 sobre $\mathbb{F}_q$. Por el Teorema 4.1.3, las raíces de $Irr(\alpha^s;\mathbb{F}_q)$ son

$$\left\{\alpha^s,\alpha^{sq},\dots,\alpha^{sq^{r-1}}\right\}=\left\{\gamma^{ds},\gamma^{dsq},\dots,\gamma^{dsq^{r-1}}\right\},$$

donde, de nuevo, r es el menor entero positivo tal que $dsq^r\equiv ds$ (mód q^t-1). Dividiendo por d, la anterior congruencia es equivalente a $sq^r\equiv s$ (mód n), pudiendo redifinir entonces la *clase q-ciclotómica de s módulo n* como el conjunto

$$C_s=\left\{s,sq,\dots,sq^{r-1}\right\}\ (\text{mód } n),$$

donde r es el entero positivo más pequeño tal que $sq^r\equiv s$ (mód n). Estas clases ciclotómicas forman una partición de $\{0,1,2,\dots,n-1\}$ y, consecuencia de la definición, $ord_n(q)$ es el cardinal de C_1. Como son disjuntas, tenemos la factorización de x^n-1 como producto de polinomios irreducibles sobre $\mathbb{F}_q$. Enunciamos todo esto en el siguiente teorema.

Teorema 4.1.6. *Sea* $n\in\mathbb{Z}^+$ *coprimo con* q*, sea* $t=ord_n(q)$ *y sea* $\alpha\in\mathbb{F}_{q^t}$ *una raíz* n*-ésima primitiva de la unidad:*

(i) Para cada $s=0,\dots,n-1$*, el polinomio irreducible de* α^s *sobre* $\mathbb{F}_q$ *es*

$$Irr(\alpha^s;\mathbb{F}_q)=\prod_{i\in C_s}(x-\alpha^i),$$

donde C_s es la clase q-ciclotómica de s módulo n.

(ii) La factorización de $x^n - 1$ en factores irreducibles sobre $\mathbb{F}_q$ es

$$x^n - 1 = \prod_s Irr(\alpha^s; \mathbb{F}_q),$$

donde s recorre un conjunto completo de representantes de las clases q-ciclotómicas módulo n.

La factorización del anterior teorema es única por ser $\mathbb{F}_q[x]$ un dominio de factorización única.

Vimos en los códigos cíclicos que un polinomio generador g de un código cíclico C en $\mathbb{F}_q^n$ es mónico y divide a $x^n - 1$. Supongamos que n y q son coprimos. Dado que en el Teorema 4.1.6 tenemos, fijada una raíz n-ésima primitiva de la unidad $\alpha \in \mathbb{F}_{q^t}$, la descomposición de $x^n - 1$ en factores irreducibles sobre $\mathbb{F}_q$, deducimos que

$$g = \prod_s Irr(\alpha^s; \mathbb{F}_q),$$

donde s recorre un subconjunto del conjunto completo de representantes de las clases q-ciclotómicas módulo n. Definimos el *conjunto de los ceros de g* como

$$Z(g) = \{\beta \in \mathbb{F}_{q^t} : g(\beta) = 0 \text{ y } \beta \text{ es una raíz } n\text{-ésima de la unidad}\}.$$

Así, fijada una raíz n-ésima primitiva de la unidad $\alpha \in \mathbb{F}_{q^t}$, es $Z(g) = \{\alpha^i : g(\alpha^i) = 0\}$

Llamaremos *conjunto de definición* de C a $T_\alpha(g) = \{i \in \mathbb{Z}_n : \alpha^i \in Z(g)\}$ y también lo denotaremos como $T_\alpha(C)$. Más adelante veremos que, como sugiere la notación, el conjunto de definición depende de la raíz n-ésima primitiva de la unidad α escogida. Diremos que un conjunto de definición T contiene s elementos *consecutivos* módulo n si hay un conjunto $\{b, b+1, \ldots, b+s-1\}$ con $s \leq n$ tal que

$$\{b, b+1, \ldots, b+s-1\} \pmod{n} \subseteq T$$

Ejemplo 4.1.7. Las clases 2-ciclotómicas módulo 7 son

$$C_0 = \{0\} \qquad C_1 = \{1, 2, 4\} \qquad C_3 = \{3, 5, 6\}$$

Si consideramos un código cíclico de longitud 7 sobre $\mathbb{F}_2$ con conjunto de definición $T = C_0 \cup C_3$, resulta que T contiene un conjunto de tres elementos consecutivos módulo 7 $\{5, 6, 0\}$.

Proposición 4.1.8. *Sea α una raíz n-ésima primitiva de la unidad en una extensión de $\mathbb{F}_q$ y sea C un código cíclico de longitud n sobre $\mathbb{F}_q$ con conjunto de definición T y polinomio generador g. Entonces todo $c \in \mathbb{F}_{q,n}[x]$ satisface la siguiente condición:*

$$c \in C \Leftrightarrow c(\alpha^i) = 0 \quad \forall i \in T$$

Demostración.

$$\begin{array}{c} c(\alpha^i)=0 \\ \forall i\in T \end{array} \Leftrightarrow \begin{array}{c} (x-\alpha^i)|c \\ \forall i\in T \end{array} \Leftrightarrow \begin{array}{c} Irr(\alpha^i;\mathbb{F}_q)|c \\ \forall i\in T \end{array} \Leftrightarrow g=\prod_{i\in T} Irr(\alpha^i;\mathbb{F}_q)|c \Leftrightarrow c\in\langle g\rangle \Leftrightarrow c\in C$$

□

4.2. Códigos BCH

El siguiente resultado es conocido y omitimos su demostración.

Lema 4.2.1. *El determinante de la matriz de Vandermonde*

$$V=\begin{pmatrix} 1 & \dots & 1 \\ a_1 & \dots & a_m \\ \vdots & \ddots & \vdots \\ (a_1)^m & \dots & (a_m)^m \end{pmatrix}$$

es $det(V)=\prod_{1\le i<j\le m}(a_j-a_i)$*, luego es no singular si los* $a_1,\dots,a_m$ *son distintos.*

Teorema 4.2.2 (Cota BCH)**.** *Sea* C *un código cíclico de longitud* n *sobre* $\mathbb{F}_q$ *con conjunto de definición* $T_\alpha(C)$ *y con distancia mínima* d*, donde* $\alpha\in\mathbb{F}_{q^t}$ *es una raíz* n*-ésima primitiva de la unidad fijada. Si* $T_\alpha(C)$ *contiene* $\delta-1$ *elementos consecutivos para algún entero* δ*, entonces* $d\ge\delta$*.*

Demostración. Sea $c=\sum_{j=1}^{w} c_{i_j}x^{i_j}$ una palabra de C de peso $w>0$ y supongamos $w<\delta$. Por la Proposición 4.1.8 es $c(\alpha^i)=0\ \ \forall i\in\{b,\dots,b+\delta-2\}$, luego

$$M\mathbf{c}^T=\begin{pmatrix} (\alpha^b)^{i_1} & \dots & (\alpha^b)^{i_w} \\ (\alpha^{b+1})^{i_1} & \dots & (\alpha^{b+1})^{i_w} \\ \vdots & \ddots & \vdots \\ (\alpha^{b+w+1})^{i_1} & \dots & (\alpha^{b+w+1})^{i_w} \end{pmatrix}\begin{pmatrix} c_{i_1} \\ \vdots \\ c_{i_w} \end{pmatrix}=0$$

y $det(M)=0$ por ser $\mathbf{c}\neq\mathbf{0}$. Ahora bien,

$$det(M)=\alpha^{(i_1+\dots+i_w)b}\begin{vmatrix} 1 & \dots & 1 \\ \alpha^{i_1} & \dots & \alpha^{i_w} \\ \vdots & \ddots & \vdots \\ (\alpha^{i_1})^{w-1} & \dots & (\alpha^{i_w})^{w-1} \end{vmatrix}$$

con esta última matriz una matriz de Valdermonde y los α^{i_j} distintos. Por el Lema anterior se tiene $det(M)\neq 0$. Contradicción. □

Ejemplo 4.2.3. En el Ejemplo 4.1.7, el conjunto de definición T contenía tres elementos consecutivos. Por tanto, el correspondiente código cíclico tendrá distancia mínima $d\ge 4$.

Fijada una raíz n-ésima primitiva de la unidad α, de la igualdad

$$g = \prod_s Irr(\alpha^s; \mathbb{F}_q)$$

y de la definición de $T_\alpha(g)$ tenemos que $deg(g) = |T_\alpha(g)|$. Así, tal y como comentamos en los códigos cíclicos (Teorema 3.3.8), $dim(C) = n - |T_\alpha(g)|$.

Definición 4.2.4. *Sea δ un entero con $2 \leq \delta \leq n$. Un código BCH*[1] *en $\mathbb{F}_q^n$ con distancia diseñada δ es un código cíclico C con conjunto de definición*

$$T = C_b \cup C_{b+1} \cup \ldots \cup C_{b+\delta-2},$$

donde cada C_i es la clase q-ciclotómica de i módulo n.
Si $b = 1$, se dice que c es un código BCH restringido.
Si $n = q^t - 1$, se dice que C es un código BCH primitivo.

De la definición se sigue que todo código BCH tiene el conjunto de elementos consecutivos

$$\{b, b+1, \ldots, b+\delta-2\} \pmod{n}.$$

Ahora bien, si C' es un código cíclico, α es una raíz n-ésima primitiva de la unidad y $b, b+1, \ldots, b+\delta-2 \in T_\alpha(C')$, tenemos que $T_\alpha(C) \subseteq T_\alpha(C')$, luego $dim(C) \geq dim(C')$ y acabamos de probar así una propiedad de los códigos BCH: tienen la máxima dimensión posible de forma que contengan el conjunto de elementos consecutivos $\{b, b+1, \ldots, b+\delta-2\}$ (mód n).

También se sigue de la definición y del Teorema 4.2.2 que todo código BCH de distancia diseñada δ tiene cota BCH δ.

Ejemplos 4.2.5.

(i) Tomando $b = 1$ y $\delta = 2$ obtenemos que todo código BCH restringido sobre $\mathbb{F}_q$ con distancia diseñada 2 es un código cíclico con conjunto de definición la clase q-ciclotómica C_1.

(ii) Busquemos un código BCH de longitud 7 sobre $\mathbb{F}_2$ con distancia diseñada 4. Las clases 2-ciclotómicas módulo 7 son:

$$C_0 = \{0\}, \qquad C_1 = \{1, 2, 4\}, \qquad C_3 = \{3, 6, 5\}$$

Tomamos $b = 0$, luego conjunto de definición $T = C_0 \cup C_1 \cup C_2 = \{0, 1, 2, 4\}$. Por la cota BCH, este código tendrá distancia mínima $d \geq 4$. Necesitamos una raíz 7-ésima de la unidad. Como $2^3 \equiv 1$ (mód 7), es $ord_7(2) = 3$ y $\mathbb{F}_{2^3} = \mathbb{F}_8$ tiene una raíz 7-ésima primitiva de la unidad. La factorización de $x^7 - 1$ en irreducibles es

$$x^7 - 1 = (x^3 + x^2 + 1)(x^3 + x + 1)(x + 1)$$

[1] Las iniciales de esta clase de códigos vienen de los matemáticos que la descubrieron de forma independiente: Raj Bose en 1960, Dijen K. Ray-Chaudhuri también en 1960 y Alexis Hocquenghem en 1959. (`https://en.wikipedia.org/wiki/BCH_code`)

Como el orden de cualquier raíz distinta de 1 es 7, entonces cualquier raíz distinta de 1 es una raíz primitiva de $\mathbb{F}_8$, luego una raíz 7-ésima primitiva de la unidad. Así, si llamamos α a una raíz de x^3+x+1, entonces

$$g=(x^3+x+1)(x+1)=1+x^2+x^3+x^4$$

es el polinomio generador de un código BCH y g tiene peso $wt(g)=4$, pero como $d\geq 4$, en este caso la cota BCH coincide con la distancia mínima del código. Es pues un $[7,3,4]$-código cíclico sobre $\mathbb{F}_2$. Anteriormente comentamos que cambiar la raíz implica cambiar el polinomio generador y éste es un ejemplo, ya que si cambiamos α por una raíz β de x^3+x^2+1, obtenemos el polinomio generador

$$g=(x^3+x^2+1)(x+1)=1+x+x^2+x^4$$

Tenemos así otro código BCH con distancia también 4 por las mismas razones.

(*iii*) Busquemos un código BCH de longitud 13 sobre $\mathbb{F}_3$ con distancia diseñada 2. Las clases 3-ciclotómicas módulo 3 son:

$$C_0=\{0\},\quad C_1=\{1,3,9\},\quad C_2=\{2,6,5\},\quad C_4=\{4,12,10\},\quad C_7=\{7,8,11\}$$

Tomamos $b=1$, luego conjunto de definición $T=C_1$. Necesitamos una raíz 13-ésima de la unidad. Tenemos $3^3\equiv 1$ (mód 13) y $ord_{13}(3)=3$, luego $\mathbb{F}_{3^3}=\mathbb{F}_{27}$ tiene una raíz 13-ésima primitiva de la unidad. La factorización de $x^{13}-1$ como producto de irreducibles es

$$x^{13}-1=(x+2)(x^3+2x+2)(x^3+x^2+2)(x^3+x^2+x+2)(x^3+2x^2+2x+1).$$

Como x^3+2x+1 es irreducible en $\mathbb{F}_3[x]$, tenemos

$$\mathbb{F}_{27}\cong\frac{\mathbb{F}_3[x]}{(x^3+2x+1)}=\left\{a_0+a_1\gamma+a_2\gamma^2 : a_0,a_1,a_2\in\mathbb{F}_3\right\}$$

con γ cumpliendo $\gamma^3+2\gamma+1=0$. De suerte que γ es una raíz primitiva de $\mathbb{F}_{27}$, por lo que $\alpha=\gamma^{(3^3-1)/13}=\gamma^2$ es una raíz 13-ésima primitiva de la unidad en $\mathbb{F}_{27}$. Tenemos $Irr(\alpha;\mathbb{F}_3)=\prod_{i\in C_1}(x-\alpha^i)$ por el Teorema 4.1.6. Calculémoslo:

$$Irr(\alpha;\mathbb{F}_3)=(x-\alpha)(x-\alpha^3)(x-\alpha^9)=-\alpha^{13}+x(\alpha^4+\alpha^{10}+\alpha^{12})+x^2(-\alpha-\alpha^3-\alpha^9)+x^3$$

Calculando todas esas potencias de α llegamos a que

$$-\alpha^{13}=2,\qquad \alpha^4+\alpha^{10}+\alpha^{12}=1\quad y\quad -\alpha-\alpha^3-\alpha^9=1.$$

Así que $g=Irr(\alpha;\mathbb{F}_3)=2+x+x^2+x^3$ es el polinomio generador de un código BCH y este código tiene distancia mínima ≥ 2 por la cota BCH, pero ≤ 4 por ser éste el peso de g. Tomemos ahora $\beta=\alpha^2$. Haciendo operaciones análogas obtenemos que $g=Irr(\beta;\mathbb{F}_3)=x^3+x^2+2$ es el polinomio generador de otro código BCH. En este caso la distancia mínima es ≥ 2 por la cota BCH y ≤ 3 por ser el peso de g. De nuevo vemos cómo, cambiando la raíz, cambiamos también el polinomio generador, pero aquí además podemos mejorar la cota BCH, ya que $Irr(\beta;\mathbb{F}_3)=Irr(\alpha^2;\mathbb{F}_3)$ y $T_{\alpha^2}(g)=\{2,6,5\}$, es decir, que "viéndolo como β" tenemos 1 elementos consecutivos (luego cota BCH 2) y "viéndolo como α^2" tenemos 2 elementos consecutivos (luego cota BCH 3). Así que el código BCH con polinomio generador x^3+x^2+2 tiene distancia mínima 3.

El siguiente teorema lo enunciamos por ser interesante, pero no lo demostraremos dado que no lo usaremos en ningún momento. Su demostración puede verse en [2].

Teorema 4.2.6. *Sea C un $[n,k]$-código BCH sobre $\mathbb{F}_q$ con distancia diseñada δ. Entonces:*

(i) $k \geq n - ord_n(q)(\delta - 1)$

(ii) Si $q = 2$ y el código es restringido, podemos suponer $\delta = 2m + 1$ impar; además, $k \geq n - ord_n(q)m$.

4.3. Decodificación de códigos BCH. Algoritmo de Sugiyama

Sea C un código BCH sobre $\mathbb{F}_q$ de longitud n y con distancia diseñada δ. Como $d(C) \geq \delta$, entonces C es, como mínimo, $t = \left\lfloor \frac{\delta-1}{2} \right\rfloor$-corrector y con este algoritmo podremos corregir hasta t errores. Supondremos para simplificar que el código BCH es restringido, por lo que será $T = \{1, \ldots, \delta - 1\}$. Sea $m = ord_n(q)$, α una raíz n-ésima primitiva de la unidad en $\mathbb{F}_{q^m}$, e $y \in \mathbb{F}_{q,n}[x]$ una palabra recibida de la que suponemos que difiere en t o menos posiciones con alguna palabra c de C. Sean $k_1, \ldots, k_v$ tales posiciones con $v \leq t$. Es entonces $y = c + e$ con

$$e = e_{k_1} x^{k_1} + \ldots + e_{k_v} x^{k_v}$$

y se trata de determinar las posiciones $k_1, \ldots, k_v$ y los valores $e_{k_1}, \ldots, e_{k_v} \in \mathbb{F}_q$. Por la Proposición 4.1.8

$$c \in C \Leftrightarrow c(\alpha^i) = 0 \quad \forall i \in T, \tag{4.1}$$

así que

$$y(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i) \quad \forall i \in \{1, \ldots, 2t\}, \tag{4.2}$$

ya que $2t \leq \delta - 1$ por ser el código $t = \left\lfloor \dfrac{\delta - 1}{2} \right\rfloor$-corrector. Definiendo

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \ldots & (\alpha^2)^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{\delta-1} & (\alpha^{\delta-1})^2 & \ldots & (\alpha^{\delta-1})^{n-1} \end{pmatrix},$$

por (4.1) podemos afirmar que

$$c \in C \Leftrightarrow c(\alpha^i) = 0 \quad \forall i \in T \Leftrightarrow H\mathbf{c}^T = 0.$$

Esto nos lleva a definir el síndrome de y como $(S_1, \ldots, S_{2t})$ donde $S_i = y(\alpha^i) \in \mathbb{F}_{q^m}$. Tenemos por (4.2) que

$$S_i = y(\alpha^i) = e(\alpha^i) = \sum_{j=1}^{v} e_{k_j} (\alpha^i)^{k_j} = \sum_{j=1}^{v} e_{k_j} (\alpha^{k_j})^i \quad \forall i \in \{1, \ldots, 2t\}$$

Para simplificar notación, denotamos

$$E_j = e_{k_j}, \qquad X_j = \alpha^{k_j} \quad \forall j = 1, \ldots, v$$

Tenemos entonces

$$S_i = \sum_{j=1}^{v} E_j X_j^i \quad \forall i \in \{1, \ldots, 2t\},$$

lo cual da lugar al sistema de ecuaciones no lineal en las X_j

$$\begin{aligned} S_1 &= E_1 X_1 + \ldots + E_v X_v \\ S_2 &= E_1 X_1^2 + \ldots + E_v X_v^2 \\ &\vdots \\ S_{2t} &= E_1 X_1^{2t} + \ldots + E_v X_v^{2t} \end{aligned} \tag{4.3}$$

con coeficientes desconocidos $E_1, \ldots, E_v$. Para lidiar con este sistema definimos el *polinomio localizador de errores* como

$$\sigma = \prod_{j=1}^{v} (1 - xX_j) \in \mathbb{F}_{q^m}[x]$$

Nótese que las raíces de este polinomio son $X_1^{-1}, \ldots, X_v^{-1}$. Por tanto, hallando σ y evaluando en los α^i $(i = 0, \ldots, n-1)$, podemos conocer sus raíces. Los exponentes de las inversas de estas raíces serán las posiciones $k_1, \ldots, k_v$ donde se están dando los errores.

Observación 4.3.1. Uno puede preguntarse en qué posición estaría el error si ocurriera $\alpha^h = \alpha^l$ con $0 \leq h, l \leq n-1$. Se tendría $\alpha^{h-l} = 1$, luego $n | h - l$. Así que sería $h - l = 0$ y $h = l$.

Una vez conozcamos $X_1, \ldots, X_v$, el sistema (4.3) nos proporciona $E_1, \ldots, E_v$. El problema reside entonces en hallar el polinomio localizador de errores σ. Para ello, definimos el *polinoimo evaluador de errores* como

$$\omega = \sum_{j=1}^{v} E_j X_j \prod_{\substack{i=1 \\ i \neq j}}^{v} (1 - xX_i) = \sum_{j=1}^{v} E_j X_j \frac{\sigma}{1 - xX_j} \in \mathbb{F}_{q^m}[x]$$

y el polinomio

$$S = \sum_{i=0}^{2t-1} S_{i+1} x^i \in \mathbb{F}_{q^m}[x],$$

donde $S_1, \ldots, S_{2t}$ son los síndromes de la palabra recibida y. Escribiendo $\frac{1}{1-xX_j}$ como la serie de potencias $\sum_{i=0}^{\infty} (xX_j)^i$, obtenemos que

$$\omega = \sigma \sum_{j=1}^{v} E_j X_j \frac{1}{1 - xX_j} = \sigma \sum_{j=1}^{v} E_j X_j \sum_{i=0}^{\infty} (xX_j)^i = \sigma \sum_{i=0}^{\infty} \left(\sum_{j=1}^{v} E_j X_j^{i+1} \right) x^i$$

$$\equiv \sigma \sum_{i=0}^{2t-1} \left(\sum_{j=1}^{v} E_j X_j^{i+1} \right) x^i \pmod{x^{2t}} \equiv \sigma \sum_{i=0}^{2t-1} S_{i+1} x^i \pmod{x^{2t}} \equiv \sigma S \pmod{x^{2t}}$$

Llamaremos *ecuación clave* a la ecuación en ω y σ

$$\omega \equiv \sigma S \ (\text{mód } x^{2t})$$

La resolución de esta ecuación nos proporcionará σ por un escalar no nulo, lo cual nos es igualmente válido para hallar las raíces de σ. Resumimos los pasos a seguir para decodificar una palabra recibida $\mathbf{y}$ y añadimos el algoritmo para resolver la ecuación clave:

Paso 1: Calcular los síndromes $S_i = y(\alpha^i) \ \ \forall i \in \{1, \ldots, 2t\}$ y S.

Paso 2: Resolver la ecuación clave $\omega = \sigma S \ (\text{mód } x^{2t})$:

Paso 2.1: Tomar $r_{-1}, r_0, b_{-1}, b_0 \in \mathbb{F}_{q^m}[x]$ con valores iniciales

$$r_{-1} = x^{2t}, \quad r_0 = S, \quad b_{-1} = 0, \quad b_0 = 1$$

Paso 2.2: Hacer las siguientes dos iteraciones hallando $h_i, r_i, b_i \in \mathbb{F}_{q^m}[x]$ para $i = 1, \ldots, l$, siendo l el índice para el cual $deg(r_{l-1}) \geq t$ y $deg(r_l) < t$:

$$r_{i-2} = r_{i-1}h_i + r_i \quad \text{con } deg(r_i) < deg(r_{i-1}) \quad (\text{dividir } r_{i-2} \text{ entre } r_{i-1})$$

$$b_i = b_{i-2} - h_i b_{i-1}$$

Paso 2.3: σ es un múltiplo de b_l por algún escalar no nulo.

Paso 3: Hallar las raíces de σ (o de b_l) mediante los cálculos $\sigma(\alpha^i)$ para $i = 0, \ldots, n-1$. Invertirlas para obtener $X_1 = \alpha^{k_1}, \ldots, X_v = \alpha^{k_v}$ y saber así las posiciones $k_1, \ldots, k_v$ en las que están los errores.

Paso 4: Resolver el sistema

$$\begin{aligned} S_1 &= E_1X_1 + \ldots + E_vX_v \\ S_2 &= E_1X_1^2 + \ldots + E_vX_v^2 \\ &\vdots \\ S_v &= E_1X_1^v + \ldots + E_vX_v^v \end{aligned}$$

para obtener $E_1 = e_{k_1}, \ldots, E_v = e_{k_v}$

Paso 5: Decodificar y por $y - e$, donde $e = e_{k_1}x^{k_1} + \ldots + e_{k_v}x^{k_v}$.

Para demostrar que el paso 2 funciona, necesitaremos probar antes dos lemas.

Lema 4.3.2. *σ y ω son coprimos.*

Demostración. Las raíces de $\sigma = \prod_{i=1}^{v}(1 - xX_i)$ son X_u^{-1} con $1 \leq u \leq v$ y

$$\omega(X_u^{-1}) = \sum_{j=1}^{v} E_jX_j \prod_{\substack{i=1 \\ i\neq j}}^{v}(1 - X_u^{-1}X_i) = E_uX_u \prod_{\substack{i=1 \\ i\neq u}}^{v}(1 - X_u^{-1}X_i) \neq 0,$$

luego no tienen factores comunes. □

Lema 4.3.3. *Sean S, h_i, r_i, b_i los polinomios del paso 2 del algoritmo de Sugiyama y sean los polinomios $f = r_{-1}$ y a_i con $i \geq -1$ satisfaciendo*

$$a_{-1} = 1 \qquad a_0 = 0 \qquad a_i = a_{i-2} - h_i a_{i-1}.$$

Entonces:

(i) $a_i f + b_i S = r_i \quad \forall i \geq -1$.

(ii) $b_i r_{i-1} - b_{i-1} r_i = (-1)^i f \quad \forall i \geq 0$.

(iii) $a_i b_{i-1} - a_{i-1} b_i = (-1)^{i+1} \quad \forall i \geq 0$*; en particular,* a_l *y* b_l *son coprimos.*

(iv) $deg(b_i) + deg(r_{i-1}) = deg(f) \quad \forall i \geq 0$.

Demostración. Probaremos todos los apartados por inducción:

$\boxed{(i)}$ $\underline{i = -1}$:

$$a_{-1} f + b_{-1} S = f = r_{-1}$$

$\underline{i = 0}$:

$$a_0 f + b_0 S = S = r_0$$

$\underline{i}$ suponiéndolo cierto para $i-1$ y $i-2$:

$$\begin{aligned} a_i f + b_i S &= (a_{i-2} - h_i a_{i-1}) f + (b_{i-2} - h_i b_{i-1}) S \\ &= a_{i-2} f + b_{i-2} S - h_i (a_{i-1} f + h_{i-1} S) \overset{\text{HI}}{=} r_{i-2} - h_i r_{i-1} = r_i \end{aligned}$$

$\boxed{(ii)}$ $\underline{i = 0}$:

$$b_0 r_{-1} - b_{-1} r_0 = r_{-1} = f$$

$\underline{i}$ suponiéndolo cierto para $i-1$:

$$\begin{aligned} b_i r_{i-1} - b_{i-1} r_i &= (b_{i-2} - h_i b_{i-1}) r_{i-1} - b_{i-1} r_i \\ &= b_{i-2} r_{i-1} - b_{i-1} (h_i r_{i-1} + r_i) \\ &= b_{i-2} r_{i-1} - b_{i-1} r_{i-2} \overset{\text{HI}}{=} -(-1)^{i-1} f = (-1)^i f \end{aligned}$$

$\boxed{(iii)}$ $\underline{i = 0}$:

$$a_0 b_{-1} - a_{-1} b_0 = -a_{-1} = -1$$

$\underline{i}$ suponiéndolo cierto para $i-1$:

$$\begin{aligned} a_i b_{i-1} - a_{i-1} b_i &= (a_{i-2} - h_i a_{i-1}) b_{i-1} - a_{i-1} (b_{i-2} - h_i b_{i-1}) \\ &= a_{i-2} b_{i-1} - a_{i-1} b_{i-2} \overset{\text{HI}}{=} -(-1)^i = (-1)^{i+1} \end{aligned}$$

$\boxed{(iv)}$ $\underline{i = 0}$:

$$deg(b_0) + deg(r_{-1}) = deg(r_{-1}) = deg(f)$$

i suponiéndolo cierto para $i-1$:

Del algoritmo de la deivisión tenemos $deg(r_i) < deg(r_{i-2})$, luego

$$deg(b_{i-1}r_i) = deg(b_{i-1}) + deg(r_i) < deg(b_{i-1}) + deg(r_{i-2}) \overset{\text{HI}}{<} deg(f)$$

y, por (ii),

$$deg(f) = deg(b_i r_{i-1}) = deg(b_i) + deg(r_{i-1})$$

□

Ya estamos en condiciones de probar que el paso 2 del algoritmo de Sugiyama funciona.

Obsérvese en primer lugar la analogía con el algoritmo de Euclides: estamos hallando el máximo común divisor de x^{2t} y S, pero "cancelamos" o "paramos" el algoritmo cuando hallamos r_l de forma que $deg(r_{l-1}) \geq t$ y $deg(r_l) < t$. ¿Y si $deg\left(mcd(x^{2t}, S)\right) \geq t$? De ser así, no existiría tal r_l. Veamos que esto no ocurre: como $wt(e) = v \leq t$, sabemos que existen σ y ω y que satisfacen $\omega \equiv \sigma S \pmod{x^{2t}}$. Entonces

$$\omega = \sigma S + ax^{2t}$$

para cierto $a \in \mathbb{F}_{q^m}[x]$ y, en consecuencia, $mcd(x^{2t}, S) | \omega$. Por tanto,

$$deg\left(mcd(x^{2t}, S)\right) \leq deg(\omega) < v \leq t$$

y existe r_l.

Ahora tenemos por (i) del Lema 4.3.3 que

$$a_l x^{2t} + b_l S = r_l, \tag{4.4}$$

luego

$$b_l S\sigma = r_l\sigma - a_l x^{2t}\sigma \tag{4.5}$$

De la ecuación clave sabemos que existe $a \in \mathbb{F}_{q^m}[x]$ de forma que

$$\omega = ax^{2t} + \sigma S, \tag{4.6}$$

por lo que

$$\sigma S b_l = \omega b_l - ax^{2t} b_l \tag{4.7}$$

De (4.5) y (4.7) deducimos que

$$r_l\sigma = \omega b_l - ax^{2t} b_l + a_l x^{2t}\sigma,$$

y que

$$r_l\sigma \equiv \omega b_l \pmod{x^{2t}}$$

Probando que $r_l\sigma$ y ωb_l tienen grado menor que $2t$, obtendremos la igualdad

$$r_l\sigma = \omega b_l, \tag{4.8}$$

Veámoslo:

- $deg(r_l\sigma) = deg(r_l) + deg(\sigma) < t + deg(\sigma) = t + v \leq 2t$
- Usando que $deg(b_l) = deg(x^{2t}) - deg(r_{l-1})$ por el Lema 4.3.3 (iv), tenemos

$$deg(\omega b_l) = deg(\omega) + deg(b_l) < deg(\omega) + t \leq v + t \leq 2t$$

Sabiendo ahora (4.8), de nuevo por (4.5) y (4.7) obtenemos

$$a_l\sigma = ab_l, \tag{4.9}$$

luego $a_l | ab_l$, y como a_l y b_l son coprimos por el Lema 4.3.3 (iii), entonces existe $\lambda \in \mathbb{F}_{q^m}[x]$ de manera que $a = \lambda a_l$, teniéndose

$$\sigma = \lambda b_l. \tag{4.10}$$

Sustituyendo en (4.6) obtenemos

$$\omega = \lambda a_l x^{2t} + \lambda b_l S \overset{(4.4)}{=} \lambda r_l$$

Si fuera $\lambda = 0$, sería $\sigma = \omega$ por (4.9) y (4.10), pero σ y ω son coprimos por el Lema 4.3.2, luego $\lambda \neq 0$ y se verifica el paso 2 de este algoritmo.

Ejemplo 4.3.4. Sea C el $[15, 7]$-código BCH restringido sobre $\mathbb{F}_2$ con distancia diseñada 5, el cual tiene conjunto de definición $T = \{1, 2, 3, 4, 6, 8, 9, 12\}$. Usando la 15-ésima raíz primitiva de la unidad α que satisface $\alpha^4 + \alpha + 1 = 0$, el polinomio generador de C es $g = 1 + x^4 + x^6 + x^7 + x^8$. Supongamos que recibimos $y = 1 + x + x^5 + x^6 + x^9 + x^{10}$. Tenemos t=$\left\lfloor \frac{\delta-1}{2} \right\rfloor = 2$ y:

$$\begin{aligned}
S_1 &= y(\alpha) = 1 + \alpha + \alpha^5 + \alpha^6 + \alpha^9 + \alpha^{10} = \alpha^2 \\
S_2 &= y(\alpha^2) = y(\alpha)^2 = \alpha^4 \\
S_3 &= y(\alpha^3) = 1 + \alpha^3 + \alpha^{15} + \alpha^{18} + \alpha^{27} + \alpha^{30} = \alpha^{11} \\
S_4 &= y(\alpha^4) = y(\alpha^2)^2 = \alpha^8 \\
S &= \alpha^2 + \alpha^4 x + \alpha^{11} x^2 + \alpha^8 x^3.
\end{aligned}$$

Resolvemos $\omega = \sigma S$ (mód x^4):

$$\begin{array}{lllllll}
 & x^4 & 0 & 0 & 0 & 0 & \underline{\left|\alpha^8 x^3 + \alpha^{11} x^2 + \alpha^4 x + \alpha^2\right.} \\
i = 1 \;\rightarrow & 0 & \alpha^3 x^3 & +\alpha^{11} x^2 & +\alpha^9 x & & \alpha^7 x + \alpha^{10} = h_1 \\
 & & 0 & \alpha x^2 & +\alpha^4 x & +\alpha^{12} = r_1 &
\end{array}$$

$$b_1 = 0 + (\alpha^7 x + \alpha^{10}) \cdot 1 = \alpha^7 x + \alpha^{10}$$

$$\begin{array}{lllll}
 & \alpha^8 x^3 & \alpha^{11} x^2 & \alpha^4 x & \alpha^2 \qquad \underline{\left|\alpha x^2 + \alpha^4 x + \alpha^{12}\right.} \\
i = 2 \;\rightarrow & 0 & 0 & 0 & \alpha^2 = r_2 \qquad \alpha^7 x = h_2
\end{array}$$

$$b_2 = 1 + \alpha^7 x(\alpha^7 x + \alpha^{10}) = \alpha^{14} x^2 + \alpha^2 x + 1$$

σ es un múltiplo de $b_2 = \alpha^{14}x^2 + \alpha^2 x + 1$ por algún escalar no nulo. Calculando $b_2(\alpha^i)$ $(i = 0, \ldots, 14)$, se tiene que $b_2(\alpha^5) = b_2(\alpha^{11}) = 0$, luego α^5 y α^{11} son las raíces de σ y sus respectivas inversas $X_1 = \alpha^{10}$ y $X_2 = \alpha^4$ nos indican que los errores están en las posiciones 10 y 4. Como el código que estamos tratando es binario y sabemos que hay errores en las posiciones 10 y 4, debe ser $E_1 = E_2 = 1$. Decodificamos y por

$$y - e = 1 + x + x^5 + x^6 + x^9 + x^{10} - (x^4 + x^{10}) = 1 + x + x^4 + x^5 + x^6 + x^9.$$

Bibliografía del capítulo

Asensio-Caruncho-Martínez [1], Cory-Pless [2], Ling-Xing [6].

Bibliografía

[1] J. Asensio, J.R. Caruncho, J. Martínez. *Ecuaciones Algebraicas*. Diego Marín, 2002.

[2] W. Cary, V. Pless *Fundamentals of Error Correcting Codes*. Cambridge University Press, 2003.

[3] Á. del Río, J.J. Simón, A. del Valle. *Álgebra Básica*. Diego Marín, 2000.

[4] J.L. García. Apuntes de la asignatura *Álgebra Conmutativa* del curso 2017/2018.

[5] D.G. Hoffman, D.A. Leonard, C.C. Lindner, K.T. Phelps, C.A. Rodger, J.R. Wall. *Coding Theory. The Essentials*. Marcel Dekker, 1991.

[6] S. Ling, C. Xing. *Coding Theory. A First Course*. Cambridge University Press, 2004.

[7] J. Martínez. Apuntes de la asignatura *Álgebra Lineal* del curso 2011/2012.

[8] J.J. Simón. Apuntes 2017. Códigos correctores de errores.

[9] Imagen de la portada: `http://www.um.es/web/matematicas/`