Universidad
de La Laguna

Introducción a la Teoría de Códigos: Códigos Cíclicos

Introduction to Coding Theory: Cyclic Codes

Raquel María Hernández Falcón

Trabajo de Fin de Grado

Departamento de Matemáticas, Estadística e Investigación Operativa

Facultad de Matemáticas

Universidad de La Laguna

La Laguna, 17 de julio de 2014

Dr. Dña. **María Victoria Reyes Sánchez**, con N.I.F. 42.040.774-V profesora Titular de Universidad adscrita al Departamento de Matemáticas, Estadística e Investigación Operativa de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

"*Introducción a la Teoría de Códigos: Códigos Cíclicos.*"

ha sido realizada bajo su dirección por Dña. **Raquel María Hernández Falcón**, con N.I.F. 54.110.041-B.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firma la presente en La Laguna a 17 de julio de 2014

Agradecimientos

A mi familia en especial mis padres y hermano,
ellos han velado por mi bienestar y educación, siendo mi mayor apoyo y depositando su entera confianza en cada reto, sin dudar de mi confianza y capacidades. Su tenacidad y lucha incansable son para mi de destacar y un ejemplo a seguir.

A mi novio,
por su esfuerzo, apoyo,tesón y sobre todo paciencia.

Al resto de mi familia y padres de mi novio,
continuamente informándose, dándome ánimos, brindándome apoyo y demostrando su orgullo por cada uno de mis logros, en especial mis abuelos.

A mi tutora Mª Victoria Reyes,"Mariví",
por su inestimable ayuda y colaboración, siendo consultora y soporte. Por su calidad humana para instruirme y guiarme,dedicándome su tiempo y brindándome sus conocimientos. Es un inmenso placer concluirlo y tener su ayuda para defenderlo con propiedad, base, entereza y la firmeza necesaria.

A mis profesores,
a quienes les debo gran parte de mis conocinientos, gracias por sus enseñanzas y su perseverancia. Y a la prestigiosa ULL que me abrió las puertas preparándome para un futuro competitivo a través de la formación.

Resumen

El objetivo de este trabajo ha sido introducirnos en el estudio de los Códigos Correctores de Errores. Debido a la dimensión que abarca este título, la memoria se ha centrado en el análisis de los Códigos Cíclicos.

En primer lugar, se estudiaron los códigos en general, su definición y sus propiedades. A su vez, fue necesario recordar algunos conceptos claves de la teoría de cuerpos finitos, como la factorización de $x^n - 1$ en polinomios irreducibles en $\mathbb{F}_q[x]$ y el cálculo de su cuerpo de descomposición, así como los elementos de éste.

A continuación, se estudiaron los códigos lineales que aparecen cuando un código forma un subespacio vectorial de $\mathbb{F}_q^n$, caracterizados por su matriz de control y matriz generadora. Cuando un código lineal es cerrado por el desplazamiento $c_0c_1...c_{n-1} \longrightarrow c_{n-1}c_0...c_{n-2}$ se dice que el código es cíclico. Dichos códigos forman el eje central de este trabajo. Se recopilaron varias formas de definirlos, propiedades y ejemplos detallados, así como los procesos precisos para codificarlos y decodificarlos, y los procedimientos de detección y corrección de errores.

El siguiente paso fue centrarnos en el análisis de los códigos BCH, que son un ejemplo de códigos cíclicos descritos a través de raíces n-ésimas de la unidad. Finalmente, llegamos a un caso particular de código BCH, los códigos Reed-Solomon, cuya longitud es $n = q - 1$ y su importancia reside en el gran número de aplicaciones tecnológicas que tiene.

La memoria incluyó la descripción simplificada del proceso de codificación y decodificación de un CD, en el que intervienen dos códigos Reed-Solomon.

El estudio realizado permitió descubrir la importancia y presencia de los códigos correctores en el mundo que nos rodea.

Palabras clave: Código Lineal, Código Cíclico, Código BCH, Código Reed-Solomon, Criptografía.

Abstract

The aim of this report has been introducing Error-Correcting Codes. Due to the dimension spanning this title, memory has focused on the analysis of cyclic codes.

First, we have studied codes in general, its definition and its properties. In turn, it was necessary to recall some key concepts of the finite fields theory, as Factoring $x^n - 1$ in irreducible polynomials in $\mathbb{F}_q[x]$ and calculation of the Splitting Field and its elements.

Right after, we studied linear codes that appear when a code forms a vectorial subspace of $\mathbb{F}_q^n$, with parity check matrix and generator matrix. When a linear code is closed by displacement $c_0c_1 \dots c_{n-1} \longrightarrow c_{n-1}c_0 \dots c_{n-2}$ its says that the code is clyclic. These codes were the core of this work. Several ways to define properties detailed examples, as well as the precise processes to encode and decode and also methods to detect and correct errors we collected.

The next step was focusing on the analysis of BCH codes, which are an example of cyclic codes defined through n-th of unity. We finally got to a particular case of BCH code, Reed-Solomon codes, whose lenght is $n = q - 1$ and its importance lies in the large number of technological applications it has.

The report included a simplified description of the process of coding and decoding of a CD in which two Reed-Solomon codes are involved. The study allowed to discover the importance and presence of the error-correcting codes in the world around us.

Keywords: *Linear code, Cyclic code, BCH code, Reed-Solomon code, Cryptography.*

Índice general

Motivación y objetivos **1**

1. Códigos Correctores de Errores. Códigos Lineales **2**
1.1. Transmisión de mensajes y Códigos lineales 2
1.2. Matriz Generadora 3
1.3. Matriz de Control 3
1.4. Distancia de Hamming y Peso 4

2. Códigos Cíclicos **5**
2.1. Definición, propiedades y ejemplos de códigos cíclicos 5
2.2. Matriz generadora y Matriz de control de un código cíclico 14
2.3. Ceros de Polinomios 17
2.4. Codificación cíclica 19
2.5. Detección y corrección de errores. Decodificación de códigos cíclicos 22

3. Códigos BCH y Códigos Reed-Solomon **25**
3.1. Definición de los códigos BCH, ejemplos y propiedades 25
3.2. Decodificación y corrección de errores en los códigos BCH 30
3.3. Códigos Reed-Solomon 33

4. Aplicación: Códigos Reed-Solomon y CD's **36**
4.1. Codificación y grabado 37
4.2. Decodificación y corrección de errores 39

5. Conclusiones **41**

A. Factorización de $x^n - 1$ sobre un cuerpo finito $\mathbb{F}_q$ **42**

Bibliografía **44**

Motivación y objetivos

Indudablemente estamos en el siglo de la información, lo cual hace que la transmisión de datos de manera segura y fiable se convierta en algo esencial y necesario. El problema está en que los canales de comunicación a través de los cuales se transfiere la información pueden conducir a errores o interferencias, o puede que sean poco seguros o fiables. Aquí entra en juego la Teoría de Códigos, que ha sido y sigue siendo un área muy activa de las matemáticas.

La Teoría de Códigos busca formas eficientes de codificar la información para que los errores mencionados puedan ser detectados e incluso corregidos. En realidad, codificar mensajes es reescribirlos de forma diferente; ya sea para evitar fallos, recuperar los datos u ocultar y proteger los mensajes (*Criptografía*).

Este campo tan interesante y actual ha motivado la elaboración de este Trabajo de Fin de Grado. Además, el uso múltiples ingredientes matemáticos como el Álgebra Lineal, la Estructuras Algebraicas o la Teoría de Cuerpos finitos permiten ver una aplicación más práctica de los conocimientos adquiridos en el Grado.

El objetivo central de esta memoria es dar una introducción a la Teoría de Códigos, concretamente a los *Códigos Correctores de Errores*. En primer lugar introduciremos nociones básicas de los códigos en general y trataremos un caso particular que son los *Códigos Lineales*. A continuación, en el segundo capítulo particularizaremos en un tipo de código lineal compuesto por los *Códigos Cíclicos*; aquí se incluyen múltiples definiciones, propiedades y ejemplos. Para casos concretos seremos capaces de contestar a la pregunta: ¿Cómo podemos saber si una información recibida ha sido distorsionada? Y si es el caso, ¿cómo podríamos recuperar los datos originales?. El tercer capítulo profundiza un ejemplo de códigos cíclicos: los *Códigos BCH* que contienen el conjunto de los *Códigos Reed-Solomon*, cuya especial importancia reside en su aplicación en mucho de los aparatos electrónicos que nos rodean. Por ello, el último capítulo se reserva a los *CD's* y cómo aparecen los códigos Reed-Solomon en ellos. Finalizamos incluyendo un apéndice donde el lector puede consultar cómo *Factorizar* $x^n - 1$ *en irreducibles sobre un cuerpo finito.*

Capítulo 1

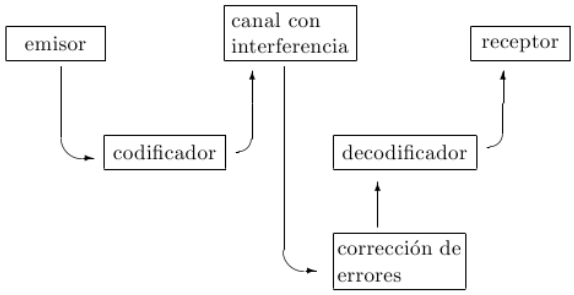
Códigos Correctores de Errores. Códigos Lineales

1.1. Transmisión de mensajes y Códigos lineales

En la actualidad la transmisión de datos es algo muy habitual aunque en numerosas ocasiones no seamos conscientes de ello. Ocurre que, en general, al enviar un mensaje o transferir información por cualquier procedimiento deseamos que el *receptor* del mensaje reciba lo mismo que el *emisor* envió; y si no es así, que sea capaz de darse cuenta que hubo errores en la transmisión e incluso, en ocasiones, corregirlos. En este contexto es donde desempeñan su papel los *Códigos Correctores de Errores* que tratan de detectar y corregir errores en los mensajes transmitidos a través de *canales*.

Los ingredientes básicos de un código son un *alfabeto* $\mathcal{A}$ o conjunto finito de símbolos, y una serie de *palabras* que no son más que sucesiones de elementos de $\mathcal{A}$ que pueden tener longitud fija (en cuyo caso hablamos de *códigos de bloque*) o variable. Así, hablamos de código q-ario si $|\mathcal{A}| = q$. Usualmente, trataremos con códigos binarios, esto es $q = 2$, y de longitud fija n.

El siguiente esquema muestra cómo sería el proceso de transmisión de un mensaje en el cual se han podido producir errores debido a *interferencias* o *ruidos* en el canal, usando códigos correctores de errores:



Dentro de los códigos correctores de errores estudiaremos los *Códigos lineales.*

Definición 1.1.1. *Un código q-ario se dice que es* ***lineal*** *de longitud fija n si es un subespacio vectorial de* $\mathbb{F}_q^n$ *(es decir, si* $\mathcal{C} \subseteq \mathbb{F}_q^n$*), donde* $\mathbb{F}_q$ *es un cuerpo finito de q elementos con q una potencia de un primo p (siendo p la carácterística de* $\mathbb{F}_q$*). Se suelen escribir como:*

$$\mathcal{C} = \{x = (x_1, x_2, ..., x_n) \ : \ x_i \in \mathbb{F}_q\}$$

Notación 1.1.1. *Si el código lineal verifica que* $dim(\mathcal{C}) = k$*, se dice que* $\mathcal{C}$ *es un* $[n, k]_q$*-código.*

En las siguientes secciones, vamos a presentar algunos conceptos esenciales para el estudio de los códigos correctores. Para un estudio más detallado ver por ejemplo [7].

1.2. Matriz Generadora

Una matriz $G \in M_{k\times n}(\mathbb{F}_q)$ es una *matriz generadora* de un código lineal $\mathcal{C}$ si sus filas forman una base del espacio vectorial $\mathcal{C}$. Se tiene así que:

$$\mathcal{C} = \mathbb{F}_q^k G = \{c = a\,G \ : \ a \in \mathbb{F}_q^k\}$$

Es decir, que podemos codificar cualquier mensaje a través de la fórmula $a\,G$. Hay que tener cuidado, porque al igual que las matrices asociadas a aplicaciones lineales no son únicas, las matrices generadoras tampoco lo son.

Cabe destacar que si la matriz G tiene rango k y presenta la forma $(I_k \mid B)$, entonces se dice que la matriz generadora es *estándar* y que la codificación es *sistemática.* Basta hacer transformaciones elementales en las filas o columnas de G para ver que todo código lineal es equivalente a un código sistemático, es decir que siempre podemos escribir:

$$a\,G = a\,(I_k \mid B) = (a \mid a\,B)$$

de forma que el mensaje aparece en las k primeras componentes de la palabra código.

1.3. Matriz de Control

Para definir la matriz de control de un código, veamos antes:

Definición 1.3.1. *Dado un código lineal* $\mathcal{C}$*, el* ***código dual*** *de* $\mathcal{C}$ *viene dado por:*

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \ / \ x\,c = 0 \ , \ \forall c \in \mathcal{C}\}$$

Es fácil ver que si G es la matriz generadora de $\mathcal{C}$ entonces:

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \ / \ x\,G^T = 0\} = \{x \in \mathbb{F}_q^n \ / \ G\,x^T = 0\}$$

Y como el rango de G es k, entonces $dim(\mathcal{C}^\perp) = n - k$.

Decimos que si $\mathcal{C}$ es un código lineal, $H \in M_{(n-k)\times n}(\mathbb{F}_q)$ es una *matriz de control o de paridad* si genera al código dual $\mathcal{C}^\perp$.

Por otro lado, se verifica que $\mathcal{C}^{\perp\perp} = \mathcal{C}$, ya que por la propia definición $\mathcal{C} \subseteq (\mathcal{C}^\perp)^\perp$ y además $dim(\mathcal{C}^{\perp\perp}) = n - (n-k) = k = dim(\mathcal{C})$. Por este motivo, la matriz generadora de $\mathcal{C}$ es la matriz de control de $\mathcal{C}^\perp$.

Asimismo, se puede dar una definición del código $\mathcal{C}$ a partir de su matriz de control:

$$\mathcal{C} = \{x \in \mathbb{F}_q^n \ : \ x\,H^T = 0\} = \{x \in \mathbb{F}_q^n \ : \ H\,x^T = 0\}$$

Y si $H \in M_{(n-k)\times n}(\mathbb{F}_q)$ es de rango máximo $n-k$ existe un único código que tiene a H por matriz de control. Además se cumple que H es una matriz de control de $\mathcal{C}$ si, y sólo si, $H \in M_{(n-k)\times n}(\mathbb{F}_q)$, $rg(H) = n-k$ y $G\cdot H^T = 0$.

Como consecuencia de la anterior relación, podemos obtener una matriz de control de un código a partir de su matriz generadora y viceversa. En particular, si $G = (I_k \mid B)$ es una matriz generadora del código en su forma estándar, entonces $H = (-B^T \mid I_{n-k})$ es la matriz de control.

1.4. Distancia de Hamming y Peso

Se denomina **distancia de Hamming** entre dos palabras de igual longitud de un código $\mathcal{C}$, al número de coordenadas distintas que poseen; y se denota por d_H. Esto es, si $x = (x_1, ..., x_n)$ y $z = (z_1, ..., z_n) \in \mathcal{C}$:

$$d_H(x,z) = \{\sharp i \ : \ x_i \neq z_i \ , \ 1 \leq i \leq n\}$$

Análogamente, se define la distancia mínima de un código como:

$$d(\mathcal{C}) = d_H(\mathcal{C}) = \text{mín}\{d_H(x,z) \ : \ x, z \in \mathcal{C} \ , x \neq z\}$$

Esto nos permite mejorar la notación 1.1.1, denotando a $\mathcal{C}$ como un $[n,k,d]_q$-código.

Por otra parte,si $x \in \mathcal{C}$, se llama **peso** de x, $w(x)$, al número de componentes no nulas de esta palabra, es decir $w(x) = d_H(x,0)$ y el peso de un código es:

$$w(\mathcal{C}) = \text{mín}\{w(x) \ : \ x \in \mathcal{C}\}$$

Proposición 1.4.1. *Si $\mathcal{C}$ es un código lineal, entonces $d(\mathcal{C}) = w(\mathcal{C})$.*

Demostración. Por una parte, como el peso es una distancia, entonces ha de ser mayor que la distancia mínima, o sea que $d(\mathcal{C}) \leq w(\mathcal{C})$.

Por otra parte, como $\mathcal{C}$ es un conjunto finito, existen x, $y \in \mathcal{C}$ tales $d(\mathcal{C}) = d(x,y)$. Además, se observa que $d(x,y) = d(x-y,0) = w(x-y)$, y $x - y \in \mathcal{C}$ por ser un código lineal; luego $w(\mathcal{C}) \leq d(x,y) = d(\mathcal{C})$. □

La distancia de un código es importante en la corrección de errores ya que si en una transmisión se emite una palabra $x \in \mathcal{C}$ y se recibe $y \notin \mathcal{C}$, el número de errores producidos es $d(x,y) = e$ y se dice que el error tiene *peso e*. Así, para decodificar se usa el criterio del *vecino más próximo*, es decir, se supone siempre que el número de errores en la transmisión es el menor posible y se decodifica por la palabra más cercana.

Capítulo 2

Códigos Cíclicos

2.1. Definición, propiedades y ejemplos de códigos cíclicos

Comencemos este capítulo dando la definición de los códigos cíclicos:

Definición 2.1.1. *Un código $[n, k, d]$ lineal $\mathcal{C}$ sobre el cuerpo $\mathbb{F}_q$ se dice que es* ***cíclico*** *si para cualquier palabra del código $c = (c_0, c_1, ..., c_{n-1})$ se tiene que $\sigma(c) = (c_{n-1}, c_0, ..., c_{n-2})$ es también una palabra del código.*

En otras palabras, un código lineal $\mathcal{C}$ es cíclico si es cerrado por el desplazamiento cíclico:

$$c_0c_1...c_{n-1} \to c_{n-1}c_0...c_{n-2}$$

Como consecuencia, $\mathcal{C}$ es un código cíclico si $\sigma^r(c) \in \mathcal{C}$ para $c \in \mathcal{C}$ y $r \in \mathbb{N}$ $(r \geq 0)$; pero esto es lo mismo que decir que $\mathcal{C}$ es un código cíclico si:

$$c_0c_1...c_{n-1} \in \mathcal{C} \Rightarrow c_r...c_{n-1}c_0...c_{r-1} \in \mathcal{C}$$

Veamos algunos ejemplos sencillos:

Ejemplo 2.1.1. *El código $\mathcal{C} = \{0\} \subset \mathbb{F}_q^n$ es un código cíclico con parámetros $[n, 0]_q$.*

Ejemplo 2.1.2. *El código de repetición, dado por:*

$$\mathcal{C} = \{(0, 0, ..., 0), (1, 1, ..., 1), ..., (q-1, q-1, ..., q-1)\} \subset \mathbb{F}_q^n$$

también es un código cíclico pero de parámetros $[n, 1]_q$.

Ejemplo 2.1.3. *El código de paridad, es decir:*

$$\mathcal{C} = \{x \in \mathbb{F}_q^n / \sum_{i=1}^{n} x_i = 0\} \subset \mathbb{F}_q^n$$

cumple también las condiciones de ser código cíclico de parámetros $[n, n-1]_q$, pues aunque permutemos el orden de los sumandos, la suma seguirá dando 0.

Ejemplo 2.1.4. *El propio $\mathbb{F}_q^n$ es un código cíclico con parámetros $[n, n]_q$.*

Ejemplo 2.1.5. *Veamos ahora en este ejemplo cómo el siguiente código binario de longitud* 7 *también es un código cíclico:*

$$\mathcal{C} = \left\{ \begin{array}{cc} 0000000 & 1111111 \\ 1101000 & 0010111 \\ 0110100 & 1001011 \\ 0011010 & 1100101 \\ 0001101 & 1110010 \\ 1000110 & 0111001 \\ 0100011 & 1011100 \\ 1010001 & 0101110 \end{array} \right\}$$

Es inmediato ver que $\mathcal{C}$ *es lineal. Y como* $\mathcal{C}$ *está formado por las palabras* $0 = 0000000$, $1 = 1111111$ *y los desplazamientos cíclicos de la palabra* $c = 1101000$ *y de su complemento* $\overline{c} = 0010111$, *se observa que* $\mathcal{C}$ *es cíclico. Sin embargo, los códigos* $\mathcal{C}_1$ *y* $\mathcal{C}_2$ *formados por la primera y la segunda columna de la matriz que define a* $\mathcal{C}$ *respectivamente, no son códigos cíclicos. El motivo principal de este hecho es que ni siquiera son códigos lineales, ya que por ejemplo la suma no es interna, tal y como se muestra a continuación:*

$$1010001 + 0100011 = 1110010$$

Observación 2.1.1. *No es cierto que si un código lineal* $\mathcal{C} \subset \mathbb{F}_q^n$ *es cerrado por algún desplazamiento cíclico* σ^k, *con* $k > 1$, *entonces* $\mathcal{C}$ *es cíclico. Por ejemplo, si* $n = 4$ *y* $k = 2$, *tenemos* $c_1c_2c_3c_4 \longrightarrow c_3c_4c_1c_2 \longrightarrow c_1c_2c_3c_4$, *y nada asegura que* $c_4c_1c_2c_3$ *esté en el código si lo está* $c_1c_2c_3c_4$; *comprobándose que este código no es cíclico pese a ser cerrado por el desplazamiento* σ^2.

A continuación, vamos a dar una caracterización de los códigos cíclicos. Para ello, en primer lugar, identificammos cada elemento del código con un polinomio de grado menor que n por medio del monomorfismo:

$$\begin{array}{rccc} \varphi: & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q[x] \\ & (c_0, c_1, ..., c_{n-1}) & \longrightarrow & c(x) := c_0 + c_1\, x + ... + c_{n-1}\, x^{n-1} \end{array}$$

Por otra parte, si $f(x)$, $g(x) \in \mathbb{F}_q[x]$ el teorema de la división euclídea establece que existen $q(x)$, $r(x) \in \mathbb{F}_q[x]$ tales que:

$$f(x) = g(x) \cdot q(x) + r(x)$$

con $deg(r(x)) < deg(g(x))$ ó $r(x) = 0$. Vamos a denotar $R(f(x), g(x)) = r(x)$, de forma que si $c(x) = c_0 + c_1\, x + ... + c_{n-1}\, x^{n-1}$, entonces:

$$\begin{array}{c} x \cdot c(x) = x\,(c_0 + c_1\, x + ... + c_{n-1}\, x^{n-1}) = c_0\, x + c_1\, x^2 + ... + c_{n-1}\, x^n = \\ = c_{n-1}\,(x^n - 1) + (c_{n-1} + c_0\, x + ... + c_{n-2}\, x^{n-1}) = c_{n-1}\,(x^n - 1) + \sigma(c)(x) \end{array}$$

Es decir que $R(x \cdot c(x), x^n - 1) = \sigma(c)(x)$; y de forma más general:

Proposición 2.1.1. *Sea $\mathcal{C}$ un código cíclico, $c(x) \in \mathcal{C}$ y $r \in \mathbb{N}$, se tiene que:*

$$\sigma^r(c(x)) = R(x^r\, c(x), x^n - 1)$$

Demostración. Para probar este resultado, vamos a usar la inducción sobre $r \in \mathbb{N}$. El caso $r = 1$ está demostrado en la proposición anterior. Suponemos que la igualdad es cierta para $r-1$ (es decir, $\sigma^{r-1}(c(x)) = R(x^{r-1}\cdot c(x), x^n - 1)$) y vamos a demostrarla para r. Con este fin, vemos que:

$$\begin{aligned} x^r\, c(x) = x\cdot x^{r-1} = x(p(x)(x^n-1) + R(x^{r-1}\, c(x), x^n - 1)) = \\ = x\, p(x)(x^n-1) + x\, \sigma^{r-1}(c(x)) \end{aligned}$$

para $p(x) \in \mathbb{F}_q[x]$. Pero como $\sigma^{r-1}(c(x)) \in \mathcal{C}$, entonces:

$$x\, \sigma^{r-1}(c(x)) = q(x)(x^n-1) + \sigma(\sigma^{r-1}(c(x))) = q(x)(x^n-1) + \sigma^r(c(x))$$

y $deg(\sigma^r(c(x))) < n$, entonces hemos llegado a que $x^r\, c(x) = (x\, p(x) + q(x))(x^n - 1) + \sigma^r(c(x))$ lo que significa que $R(x^r\cdot c(x), x^n - 1) = \sigma^r(c(x))$. □

Como consecuencia inmediata, tenemos el siguiente resultado:

Proposición 2.1.2. *Un código lineal $\mathcal{C} \subseteq \mathbb{F}_q[x]$ es cíclico si y sólo si para cualquier $c(x) \in \mathcal{C}$ y para cualquier $r \geq 0$ se cumple que $R(x^r\cdot c(x), x^n - 1) \in \mathcal{C}$.*

Este resultado sugiere que el contexto natural en el que se deben estudiar los código cíclicos es el anillo cociente:

$$A_{q,n} := \frac{\mathbb{F}_q[x]}{(x^n - 1)}$$

cuyos elementos son las clases de equivalencia definidas por la relación:

$$f(x) \equiv g(x)\ (mod\ x^n - 1) \Longleftrightarrow R(f(x), x^n - 1) = R(g(x), x^n - 1)$$

El anillo $A_{q,n}$ es una $\mathbb{F}_q$-álgebra y la composición:

$$\begin{array}{rccccc} \phi : \mathbb{F}_q^n & \xrightarrow{\varphi} & \mathbb{F}_q[x] & \xrightarrow{\pi} & A_{q,n} \\ c & \to & c(x) & \to & c(x) + (x^n - 1) \end{array}$$

es un isomosrfismo de $\mathbb{F}_q$-espacios vectoriales. En particular, si $\mathcal{C} \subseteq \mathbb{F}_q^n$ es un código lineal podemos identificar $\mathcal{C} \cong \phi(\mathcal{C})$ y los códigos cíclicos van a ser precisamente los ideales de $A_{q,n}$ como veremos a continuación. Para ello, previamente probamos el siguiente lema.

Lema 2.1.1. *Si $I \subseteq A_{q,n}$ es un subespacio vectorial, entonces I es un ideal si, y sólo si, para cualquier $f(x)+(x^n-1) \in I$ y para cualquier $r \geq 0$ se cumple que $x^r\, f(x)+(x^n-1) \in I$.*

Demostración. Está claro que si I es un ideal se cumple la condición. y, recíprocamente, como I es un subespacio vectorial entonces es cerrado para la suma y el producto escalar. Además, por la hipótesis, si $g(x) + (x^n - 1) = (a_0 + ... + a_{n-1}\, x^{n-1}) + (x^n - 1) \in A_{q,n}$, entonces:

$$(g(x) + (x^n - 1))(f(x) + (x^n - 1)) = g(x)\ f(x) + (x^n - 1) =$$

$$= a_0(f(x) + (x^n - 1)) + a_1(x\, f(x) + (x^n - 1)) + ... + a_{n-1}(x^{n-1}\, f(x) + (x^n - 1)) \in I$$

□

Todas estas consideraciones nos conducen a la siguiente caraterización para los códigos cíclicos:

Corolario 2.1.1. *Sea $\mathcal{C} \subseteq \mathbb{F}_q^n$ un código lineal. $\mathcal{C}$ es un código cíclico si y sólo si $\phi(\mathcal{C})$ es un ideal del anillo $A_{q,n}$.*

Demostración. Teniendo en cuenta el lema 2.1.1 y la proposición 2.1.2, entonces $\mathcal{C} \subseteq \mathbb{F}_q^n$ es un código cíclico si, y sólo si, para cualquier $c(x) \in \mathcal{C}, r \geq 0$, $R(x^r\, c(x), x^n - 1) + (x^n - 1) = x^r\, c(x) + (x^n - 1) \in \phi(\mathcal{C})$. □

En adelante, para simplificar la notación escribiremmos los elementos de $A_{q,n}$ como polinomios de grado menor que n, con el producto módulo $x^n - 1$, es decir, reemplazando los términos $a\, x^{n\,i+j}$ con $0 \leq j < n$ por $a\, x^j$.

Teorema 2.1.1. *Si $\mathcal{C} \subseteq A_{q,n} = \frac{\mathbb{F}_q[x]}{(x^n-1)}$ es un código cíclico de longitud n, entonces:*

1. *Exite un único polinomio mónico de grado mínimo $g(x)$ en $\mathcal{C}$, y este polinomio genera a $\mathcal{C}$, es decir, $\mathcal{C} =< g(x) >= \{g(x) \cdot a(x) \,/\, a(x) \in A_{q,n}\}$. Además, $g(x) | x^n - 1$.*

2. *Si $deg(g(x)) = r$, entonces:*

$$\mathcal{C} = \{u(x) \cdot g(x); deg(u(x)) < n - r\}$$

3. *El conjunto $\{g(x), x \cdot g(x), ..., x^{n-r+1} \cdot g(x)\}$ forma una base de $\mathcal{C}$ y, por lo tanto, $\mathcal{C}$ tiene dimensión $n - r$.*

Demostración. Probemos por separado los distintos apartados del teorema:

1. *a)* Vamos a ver, en primer lugar, la existencia del polinomio $g(x)$. Con este fin, sea $S = \{deg(g(x))/g(x) \in \mathcal{C}\} \subseteq \mathbb{N}$. Como $S \neq \emptyset$ y $S \subseteq \mathbb{N}$, entonces existe un primer elemento $r \geq 0$ de forma que:

$$g_1(x) = a_0 + a_1\, x + ... + a_r\, x^r \in \mathbb{C},\ a_r \neq 0$$

Por lo tanto, basta considerar $g(x) = a_r^{-1} \cdot g_1(x)$ que es mónico y está en $\mathcal{C}$.

b) Ahora, veamos la unicidad. Supongamos que $\mathcal{C}$ contiene dos polinomios mónicos $g_1(x)$ y $g_2(x)$ de grado mínimo r. Ocurre que, como $g_1(x) \neq g_2(x)$, entonces $g_1(x) - g_2(x) \in \mathcal{C} \setminus \{0\}$, al ser $\mathcal{C}$ un ideal. Asimismo, $deg(g_1(x) - g_2(x)) < r$ y esto contradice el carácter minimal de $g_1(x)$ y $g_2(x)$; concluyendo así que $g_1(x) = g_2(x)$.

c) Demostremos, por otra parte, que $g(x)$ genera a $\mathcal{C}$, es decir que $\mathcal{C} =< g(x) >$. Por doble inclusión:

$\boxed{\subseteq}$ Sea $c(x) \in \mathcal{C}$. De forma que tenemos $c(x) = c(x) + I$ y $g(x) = g(x) + I$. En $\mathbb{F}_q[x]$ podemos afirmar por el algoritmo de la división que existe $q(x) \in \mathbb{F}_q[x]$ tal que:

$$c(x) = g(x)\cdot q(x) + r(x)$$

con $deg(r(x)) < deg(g(x))$ ó $r(x) = 0$. Así pues, $r(x) = c(x) - g(x)\cdot q(x) \in \mathcal{C}$ con $deg(r(x)) < deg(g(x))$. Por tanto, $r(x) = 0$, lo cual implica que $c(x) \in < g(x) >$. Esto significa que $\mathcal{C} \subseteq < g(x) >$.

$\boxed{\supseteq}$ Sabemos que $< g(x) >= \{a(x)\cdot g(x)/a(x) \in A_{q,n}\}$. Y al ser $\mathcal{C}$ un ideal de $A_{q,n}$, si $g(x) \in \mathcal{C}$, entonces $< g(x) > \subseteq \mathcal{C}$.

d) Por último, veamos que $g(x)$ es un divisor de $x^n - 1$. Con este fin, usando de nuevo la división euclídea en $\mathbb{F}_q[x]$ esta vez para $x^n - 1$, tenemos que:

$$x^n - 1 = g(x)\cdot h(x) + r(x),\ deg(r(x)) < deg(g(x))\ o\ r(x) = 0$$

De manera que en $A_{q,n}$ tendríamos que:

$$0 = g(x)\cdot h(x) + r(x) \Longrightarrow r(x) = -g(x)\cdot h(x) \Longrightarrow r(x) \in \mathcal{C}$$

Y como $deg(r(x)) < deg(g(x))$, necesariamente $r(x) = 0$ para que no haya contradicción, por lo que $x^n - 1 = g(x)\cdot h(x)$. Y de este modo queda demostrado que $g(x) | x^n - 1$.

2. Ya probamos que $\mathcal{C} =< g(x) >$; por consiguiente,

$$\mathcal{C} = \{g(x)\cdot f(x); f(x) \in A_{q,n}\}$$

Por ello, basta ver que podemos retringirnos a $f(x)$ con $deg(f(x)) < n - r$. Sabemos que $x^n - 1 = h(x)\cdot g(x)$ para algún polinomio $h(x)$ de grado $n - r$. Ahora, dividiendo en $\mathbb{F}_q[x]$, tenemos que $f(x) = q(x)\cdot h(x) + u(x)$, con $deg(u(x)) < n - r$ ó $u(x) = 0$. Entonces,

$$f(x)\cdot g(x) = q(x)\cdot h(x)\cdot g(x) + u(x)\cdot g(x) = q(x)\cdot (x^n - 1) + u(x)\cdot g(x) \Rightarrow$$

$$\Rightarrow f(x)\cdot g(x) = u(x)\cdot g(x)\ en\ A_{q,n}$$

Esto ya prueba que, efectivamente, $\mathcal{C} = \{u(x)\cdot g(x); deg(u(x)) < n - r\}$.

3. Por el apartado anterior, queda probado que $\{g(x), x\cdot g(x), ..., x^{n-r+1}\cdot g(x)\}$ es un sistema generador de $\mathcal{C}$ con $n - r$ elementos y, además, es linealmente independiente (pues los polinomios tienen distintos grados), así que resulta evidente que se trata de una base de $\mathcal{C}$ y que $dim(\mathcal{C}) = n - r$ como bien enuncia el teorema.

□

Observación 2.1.2. *El código puede tener más generadores, pero sólo uno mónico de grado mínimo.*

Ejemplo 2.1.6. *Sean* $A_{2,3} = \frac{\mathbb{F}_2[x]}{(x^3-1)}$ *y* $g(x) = x+1 \in \mathbb{F}_2[x]$. *El código* $C =< g(x) >$ *viene dado por:*

$$\begin{aligned} \mathcal{C} =< g(x) >= \{g(x)\cdot u(x)/u(x) = ax+b; a,b \in \mathbb{F}_2\} = \\ = \{(x+1)\cdot 0, (x+1)\cdot 1, (x+1)\cdot x, (x+1)\cdot (x+1)\} = \\ = \{0, x+1, x^2+x, x^2+1\} = \{000, 011, 110, 101\} \end{aligned}$$

Se observa que $dim(\mathcal{C}) = 3-1 = 2$ *y que* $\mathcal{C}$ *tiene* $2^2 = 4$ *palabras. Sin embargo, en este ejemplo podemos comprbar que* $\mathcal{C}$ *está también generado por el polinomio* x^2+1, *dado que:*

$$\begin{aligned} < x^2+1 >= \{(x^2+1)\cdot 0, (x^2+1)\cdot 1, (x^2+1)\cdot x, (x^2+1)\cdot (x+1)\} = \\ = \{0, x^2+1, x^3+x, x^3+x^2+x+1\} = \{0, x^2+1, x+1, x^2+x\} = \mathcal{C} \end{aligned}$$

Pero x^2+1 *no es un divisor de* x^3-1 *en* $\mathbb{F}_2[x]$. *De hecho,* $x^3-1 = (x^2+x+1)(x+1)$. *Además,* x^2+1 *no es de grado mínimo, puesto que* $deg(x+1) < deg(x^2+1)$.

Definición 2.1.2. *El polinomio* $g(x)$ *descrito en el teorema anterior se conoce como* ***polinomio generador de*** $\mathcal{C}$ *y escribimos* $\mathcal{C} =\ll g(x) \gg$.

Definición 2.1.3. *Si* $g(x)$ *es el polinomio generador de un código cíclico* $\mathcal{C}$, $h(x) \in \mathbb{F}_q[x]$ *y* $g(x)\cdot h(x) = x^n - 1$, *entonces* $h(x)$ *es el* ***polinomio de control de*** $\mathcal{C}$.

Así, una condición necesaria para que $g(x)$ sea el polinomio generador de un código cíclico de longitud n es que $g(x)$ divida a $x^n - 1$. Si a eso añadimos que $g(x)$ sea mónico, entonces vale la recíproca.

Proposición 2.1.3. *Un polinomio mónico* $p(x) \in A_{q,n}$ *es el polinomio generador de un código cíclico de longitud* n *sobre* $\mathbb{F}_q$ *si y sólo si* $p(x)|x^n - 1$.

Demostración. Como ya hemos hecho con anterioridad, analicemos la doble implicación:

$\boxed{\Longrightarrow}$ La implicación hacia este lado es consecuencia del teorema 2.1.1.

$\boxed{\Longleftarrow}$ Para el recíproco, si $\mathcal{C} =< p(x) >$, vamos a ver que $p(x)$ es de grado mínimo. Supongamos que $p(x)|x^n - 1$ y que $g(x)$ es el polinomio generador de $\mathcal{C} =< p(x) >$, con $p(x) \neq g(x)$. Como $p(x)$ y $g(x)$ son mónicos, $deg(g(x)) \leq deg(p(x))$, pues $g(x)$ es de grado mínimo. Por hipótesis, $x^n - 1 = p(x)\cdot f(x)$ para algún polinomio $f(x) \neq 0$.

Por otra parte, como $g(x) \in \mathcal{C} \Rightarrow g(x) \in < p(x) > \Rightarrow \exists a(x) \in A_{q,n}$ tal que $g(x) = a(x)\cdot p(x)$. De manera que:

$$g(x)\cdot f(x) = a(x)\cdot p(x)\cdot f(x) = a(x)\cdot (x^n - 1)$$

Por lo tanto, $deg(g(x)\cdot f(x)) \geq deg(x^n - 1) = deg(p(x)\cdot f(x))$, esto quiere decir que $deg(g(x)) \geq deg(p(x))$. De este modo, hemos llegado a que $deg(g(x)) = deg(p(x))$ y, como ambos polinomios son mónicos, tienen que ser iguales; es decir, $p(x) = g(x)$. Luego, $p(x)$ es el polinomio generador de $\mathcal{C}$. □

Este último teorema, nos lleva a deducir la existencia de una correspondencia biyectiva entre el conjunto de polinomios mónicos de $\mathbb{F}_q[x]$ que dividen a x^n-1 y los códigos cíclicos en $\mathbb{F}_q^n$. Esto es:

$$\begin{array}{rccc} \psi: & D_n & \longrightarrow & I_n \\ & g(x) & \longrightarrow & \ll g(x) \gg \end{array}$$

donde $D_n = \{$ divisores mónicos de x^n-1 en $\mathbb{F}_q[x]\}$ e $I_n = \{$ ideales de $A_{q,n}\} = \{$ códigos cíclicos en $\mathbb{F}_q^n\}$.

Corolario 2.1.1. *Hay tantos códigos cíclicos de longitud n sobre $\mathbb{F}_q$ como divisores mónicos del polinomio x^n-1 en $\mathbb{F}_q[x]$.*

Esto muestra la importancia que tiene poder factorizar x^n-1 en irreducibles mónicos sobre cuerpos finitos (Ver apéndice A); puesto que si sabemos factorizar x^n-1 podemos calcular todos los códigos cíclicos y si $x^n-1 = m_1(x)\cdot m_2(x)\cdot ...m_\nu(x)$ es la descomposición de x^n-1 en irreducibles mónicos sobre $\mathbb{F}_q[x]$, entonces cada subconjunto $S \subset \{1,2,...,\nu\}$ define un código cíclico de longitud n sobre $\mathbb{F}_q$, dado por:

$$\mathcal{C}_S = \ll g_S(x) \gg \qquad \textit{donde} \qquad g_S(x) = \prod_{i\in S} m_i(x)$$

Se considera que $g_\phi(x) = 1$. Luego, hay 2^ν códigos cíclicos q-arios de longitud n. No obstante, puede suceder que algunos de éstos sean equivalentes entre sí.

Observación 2.1.3. *Con el fin de evitar que el polinomio generador de un código cíclico tenga raíces repetidas, vamos a suponer que n y q son coprimos. En este caso, x^n-1 es separable en $\mathbb{F}_q[x]$.*

Ejemplo 2.1.7. *Vamos a ver a lo largo de este ejemplo, cómo encontrar todos los códigos cíclicos binarios de longitud 7 (es decir, en $\mathbb{F}_2^7$). Para ello, como ya hemos visto, tenemos que calcular los divisores mónicos de $x^7-1 \in \mathbb{F}_2[x]$. Si vamos al ejemplo A.1, se ve que la factorización de x^7-1 en $\mathbb{F}_2[x]$ es:*

$$x^7-1 = (x+1)(x^3+x+1)(x^3+x^2+1) = p_1(x)\cdot p_2(x)\cdot p_3(x)$$

siendo $p_1(x), p_2(x)$ y $p_3(x)$ polinomios irreducibles en $\mathbb{F}_2[x]$. De esta manera, los divisores de x^7-1 son:

- 1
- $p_1(x) = x+1$
- $p_2(x) = x^3+x+1$
- $p_2(x) = x^3+x^2+1$
- $p_1(x)\cdot p_2(x) = (x+1)(x^3+x+1) = x^4+x^3+x^2+1$
- $p_1(x)\cdot p_3(x) = (x+1)(x^3+x^2+1) = x^4+x^2+x+1$

- $p_2(x)\cdot p_3(x) = (x^3+x+1)(x^3+x^2+1) = x^6+x^5+x^4+x^3+x^2+x+1$
- $p_1(x)\cdot p_2(x)\cdot p_3(x) = (x+1)(x^3+x+1)(x^3+x^2+1) = x^7-1$

Y, por lo tanto, tenemos los códigos:

- $\mathcal{C}_1 = \mathbb{F}_2^7$ *con parámetros* $[7,7]_2$
- $\mathcal{C}_2 = \ll x+1 \gg$ *con parámetros* $[7,6]_2$
- $\mathcal{C}_3 = \ll x^3+x+1 \gg$ *con parámetros* $[7,4]_2$
- $\mathcal{C}_4 = \ll x^3+x^2+1 \gg$ *con parámetros* $[7,4]_2$
- $\mathcal{C}_5 = \ll x^4+x^3+x^2+1 \gg$ *con parámetros* $[7,3]_2$
- $\mathcal{C}_6 = \ll x^4+x^2+x+1 \gg$ *con parámetros* $[7,3]_2$
- $\mathcal{C}_7 = \ll x^6+x^5+x^4+x^3+x^2+x+1 \gg$ *con parámetros* $[7,1]_2$
- $\mathcal{C}_8 = \{0\}$ *con parámetros* $[7,0]_2$

Por ejemplo, escribamos completamente el código:

$$\mathcal{C} = \ll x^3+x+1 \gg = \{(x^3+x+1)\cdot u(x)/deg(u(x)) < 4\} =$$
$$= \{(x^3+x+1)(a+bx+cx^2+dx^3)/a,b,c,d \in \mathbb{F}_2\}$$

Por lo tanto,

$u(x)$	$u(x)\cdot(x^3+x+1)$	**Palabra del Código**
0	0	0000000
1	x^3+x+1	1101000
x	x^4+x^2+x	0110100
x^2	$x^5+x^3+x^2$	0011010
x^3	$x^6+x^4+x^3$	0001101
$x+1$	$x^4+x^3+x^2+1$	1011100
x^2+1	x^5+x^2+x+1	1110010
x^3+1	x^6+x^4+x+1	1100101
x^2+x	$x^5+x^4+x^3+x$	0101110
x^3+x	$x^6+x^3+x^2+x$	0111001
x^3+x^2	$x^6+x^5+x^4+x^2$	0010111
x^3+x^2+x	x^6+x^5+x	0100011
x^3+x^2+1	$x^6+x^5+x^4+x^3+x^2+x+1$	1111111
x^3+x+1	x^6+x^2+1	1010001
x^2+x+1	x^5+x^4+1	1000110
x^3+x^2+x+1	$x^6+x^5+x^3+1$	1001011

De este modo, podemos observar que este código coincide con el propuesto en el ejemplo 2.1.5, donde ya habíamos visto a mano que se trataba de un código cíclico. La siguiente proposición muestra algunas propiedades de este tipo de códigos.

Proposición 2.1.4. *Sean* $\mathcal{C}_1 = \ll g_1(x) \gg$ *y* $\mathcal{C}_2 = \ll g_2(x) \gg$ *códigos cíclicos en* $A_{q,n}$*, entonces:*

1. $\mathcal{C}_1 \subseteq \mathcal{C}_2 \Leftrightarrow g_2(x)|g_1(x)$
2. $\mathcal{C}_1 \cap \mathcal{C}_2 = \ll m.c.m\{g_1(x), g_2(x)\} \gg$

Demostración. 1. Sea $deg(g_1(x)) = s < n$ y $deg(g_2(x)) = r < n$. Se observa que, como $g_1(x) \in \mathcal{C}_1 \subseteq \mathcal{C}_2$, entonces existe $h(x) \in A_{q,n}$ con $deg(h(x)) < n - r$ tal que $g_1(x) = h(x)\, g_2(x)$. Esto demuestra que $g_2(x)|g_1(x)$ en $\mathbb{F}_q[x]$.

Recíprocamente, si $g_2(x)|g_1(x)$ entonces existe un $h(x) \in \mathbb{F}_q[x]$ tal que $g_1(x) = h(x)\, g_2(x)$. Por lo tanto, si denotamos $deg(h(x)) = m$ tenemos que $deg(g_1(x)) = deg(h(x)) + deg(g_2(x)) = m + r = s < n$; luego, $deg(h(x)) < n - r$ y $g_1(x) = h(x)\, g_2(x)$, lo que implica que $g_1(x) \in \mathcal{C}_2$, es decir, $\ll g_1(x) \gg \subseteq \mathcal{C}_2$, y por consiguiente $\mathcal{C}_1 \subseteq \mathcal{C}_2$.

2. En primer lugar, sabemos que:

$$\left.\begin{array}{l} g_1(x)|x^n - 1 \\ g_2(x)|x^n - 1 \end{array}\right\} \Longrightarrow m.c.m(g_1(x), g_2(x))|x^n - 1$$

Ahora, llamemos $m(x) = m.c.m(g_1(x), g_2(x))$, que es mónico y divide a $x^n - 1$, y demostremos en segundo lugar que $\ll m(x) \gg = \ll g_1(x) \gg \cap \ll g_2(x) \gg$. Para ello, sea $\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$, es fácil ver que $\mathcal{C}$ es un código cíclico por lo que será de la forma $\mathcal{C} = \ll g(x) \gg = \ll g_1(x) \gg \cap \ll g_2(x) \gg$. Ahora bien:

$$\left.\begin{array}{rcl} \mathcal{C} \subseteq \mathcal{C}_1 & \Longrightarrow & g_1(x)|g(x) \\ \mathcal{C} \subseteq \mathcal{C}_2 & \Longrightarrow & g_2(x)|g(x) \end{array}\right\} \Longrightarrow m(x)|g(x) \Longrightarrow \ll m(x) \gg \subseteq \ll g(x) \gg$$

Recíprocamente:

$$\left.\begin{array}{l} m(x)|g_1(x) \\ m(x)|g_2(x) \end{array}\right\} \Longrightarrow \left.\begin{array}{l} \ll g_1(x) \gg \subseteq \ll m(x) \gg \\ \ll g_2(x) \gg \subseteq \ll m(x) \gg \end{array}\right\} \Longrightarrow$$

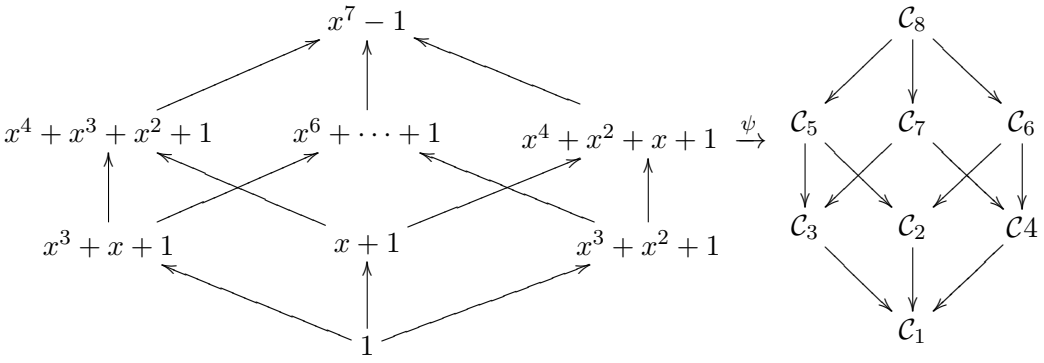
$$\Longrightarrow \ll g_1(x) \gg \cap \ll g_2(x) \gg = \ll g(x) \gg \subseteq \ll m(x) \gg$$

Y siendo ambos generadores de $\mathcal{C}$, necesariamente tienen que ser iguales. Demostrando así que $\mathcal{C}_1 \cap \mathcal{C}_2 = \ll m(x) \gg$.

□

Este resultado viene a decir que $\psi : g(x) \longrightarrow \ll g(x) \gg$ es un anti-isomorfismo de reticulados que invierte el orden entre los conjuntos $(D_n, |)$ y $(I_n, \subset)$.

Ejemplo 2.1.8. *Completando el ejemplo 2.1.7, veamos los reticulados* $(D_7, |)$ *de divisores mónicos de* $x^7 - 1$ *sobre* $\mathbb{F}_2$*, e* $(I_7, \subset)$ *de códigos cíclicos de* $A_{2,7}$*:*



2.2. Matriz generadora y Matriz de control de un código cíclico

Si $\mathcal{C} = \ll g(x) \gg$ es un código cíclico y $dim(\mathcal{C}) = k$, entonces ya vimos en el teorema 2.1.1 que una base es:

$$\mathcal{C} = \{g(x), x \cdot g(x), ..., x^{n-r-1} \cdot g(x)\}$$

Por lo tanto, si $g = g_0 + g_1 \cdot x + ... + g_{r-1} \cdot x^{r-1} + x^r$, entonces una matriz generadora es:

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & ... & g_{r-1} & 1 & 0 & 0 & ... & 0 \\ 0 & g_0 & g_1 & ... & g_{r-2} & g_{r-1} & 1 & 0 & ... & 0 \\ 0 & 0 & g_0 & ... & g_{r-3} & g_{r-2} & g_{r-1} & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & ... & & ... & g_{r-3} & g_{r-2} & g_{r-1} & 1 \end{pmatrix} \in M_{k \times n}(\mathbb{F}_q)$$

Se observa que cada fila es un desplazamiento cíclico de la anterior.

Además, si $g_0 = 0$ entonces $g(x) = g_1 \cdot x + ... + g_{r-1} \cdot x^{r-1} + x^r = x \cdot (g_1 + ... + g_{r-1} \cdot x^{r-2} + x^{r-1}) = x \cdot g'(x)$ con $deg(g'(x)) = r - 1 < r$ pero $g'(x) = 1 \cdot g'(x) = x^n \cdot g'(x) = x^{n-1} \cdot x \cdot g'(x) = x^{n-1} \cdot g(x) \in \mathcal{C}$, así que esto es absurdo pues $g'(x) \neq 0$ y tiene grado menor que $g(x)$. Luego, $g_0 \neq 0$.

Observación 2.2.1. *La matriz generadora antes descrita NO es estándar, lo que hará más difíciles las operaciones de codificación y decodificación.*

A raíz de la observación anterior y con el fin de poder realizar una codificación sistemática, vamos a considerar los k polinomios linealmente independientes siguientes:

$$r_i(x) = R(x^i, g(x)) = r_{i,0} + r_{i,1}\, x + ... + r_{i,n-k-1}\, x^{n-k-1}$$

para todo $i = n-k, ..., n-1$. Esto implica que $x^i = g(x) \cdot q_i(x) + r_i(x)$ y $deg(r_i) < r = n-k$; y por lo tanto $x^i - r_i(x) \in \mathcal{C}$ y $deg(x^i - r_i(x)) = i$, o sea que $\{x^{n-k} - r_{n-k}(x), ..., x^{n-1} -$

$r_{n-1}(x)\} \subset \mathcal{C}$ con k elementos. Estas condiciones nos permiten ver que estos polinomios conforman una base del código $\mathcal{C}$. De este modo, la matriz generadora dada por esta base es:

$$G = \left(\begin{array}{cccccccc} -r_{n-k,0} & ... & -r_{n-k,n-k-1} & 1 & 0 & ... & 0 \\ -r_{n-k+1,0} & ... & -r_{n-k+1,n-k-1} & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -r_{n-1,0} & ... & -r_{n-1,n-k-1} & 0 & 0 & ... & 1 \end{array}\right) = \left(\begin{array}{c|c} -r_{n-k}(x) & \\ -r_{n-k+1}(x) & \\ \vdots & I_k \\ -r_{n-1}(x) & \end{array}\right)$$

Notemos que esta matriz cumple que para todo $a = (a_0, ..., a_{k-1}) \in \mathbb{F}_q^k$ se tiene que $a\,G = (a_k, ..., a_{n-1}, a_0, ..., a_{k-1})$. De forma que el mensaje aparece en las últimas k posiciones de la palabra codificada y los símbolos de control en las $n-k$ primeras. Por ello, tomaremos esta matriz como matriz estándar de un código cíclico.

Teorema 2.2.1. *Si $h(x) = h_0 + h_1\,x + ... + h_{n-r}\,x^{n-r}$ es el polinomio de control de un código cíclico $\mathcal{C}$, entonces:*

1. $\mathcal{C} = \{p(x) \in A_{q,n} : h(x)\cdot p(x) = 0\}$
2. *La matriz:*

$$H = \left(\begin{array}{cccccccc} h_{n-r} & h_{n-r-1} & ... & h_0 & 0 & 0 & ... & 0 \\ 0 & h_{n-r} & ... & h_1 & h_0 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & ... & 0 & 0 & h_{n-r} & ... & h_0 \end{array}\right) \in M_{(n-k)\times n}(\mathbb{F}_q)$$

   es la matriz de control de $\mathcal{C}$.

3. *El código dual es el código cíclico de dimensión r cuyo polinomio generador es:*

$$g^{\perp}(x) = h_0^{-1}\,x^{n-r}\,h(x^{-1}) = h_0^{-1}\,(h_0\,x^{n-r} + h_1\,x^{n-r-1} + ... + h_{n-r})$$

Demostración. 1. Sea $\mathcal{C} = \ll g(x) \gg$. Entonces, si $p(x) \in \mathcal{C}$ tenemos que $p(x) = f(x)\cdot g(x)$ para algún polinomio $f(x) \in A_{q,n}$. De esta manera, ocurre que $p(x)\cdot h(x) = f(x)\cdot g(x)\cdot h(x) = f(x)(x^n - 1) = 0$.

Recíprocamente, si $p(x) \in A_{q,n}$ tal que $p(x)\cdot h(x) = 0$, usando el algoritmo de la división en $\mathbb{F}_q[x]$, se tiene que $p(x) = q(x)\cdot g(x) + r(x)$, con $deg(r(x)) < r$ o $r(x) = 0$. Entonces,

$$p(x)\cdot h(x) = q(x)\cdot g(x)\cdot h(x) + r(x)\cdot h(x) = 0 + r(x)\cdot h(x) = 0$$

y $deg(r(x)\cdot h(x)) < r + (n-r) = n$, luego $r(x)\cdot h(x) = 0$; así que $r(x) = 0$ y, por lo tanto, $p(x) = q(x)\cdot g(x)$, es decir que $p(x) \in \mathcal{C}$.

2. Si $c(x) \in \mathcal{C}$, entonces por definición $c(x)\cdot h(x) = 0$ con $deg(c(x)) < n$ y $deg(h(x)) = n-r$. Esto implica que $deg(c(x)\cdot h(x)) < n+n-r = 2n-r$. En $\mathbb{F}_q[x]$ se tendría que $c(x)\cdot h(x) = (x^n-1)\cdot q(x)$ y $deg(q(x)) < n-r$, esto es, $c(x)\cdot h(x) = (x^n-1)(q_0 + q_1\,x + ... + q_{n-r-1}\,x^{n-r-1}) = q_0\,x^n + q_1\,x^{n+1} + ... + q_{n-r-1}\,x^{2n-r-1} + (q_0 + q_1\,x + ... + q_{n-r-1}\,x^{n-r-1})$. Esto conlleva que los coeficientes de x^{n-r}, x^{n-r+1}, ..., x^{n-1} en el producto $c(x)\cdot h(x)$ son 0, es decir, tenemos el siguiente sistema de ecuaciones:

$$\left.\begin{array}{r} c_0\cdot h_{n-r} + c_1\cdot h_{n-r-1} + ... + c_{n-r}\cdot h_0 = 0 \\ c_1\cdot h_{n-r} + c_2\cdot h_{n-r-1} + ... + c_{n-r+1}\cdot h_0 = 0 \\ \vdots \\ c_{r-1}\cdot h_{n-r} + c_r\cdot h_{n-r-1} + ... + c_{n-1}\cdot h_0 = 0 \end{array}\right\}$$

Esto es equivalente a decir que $(c_0c_1...c_{n-1})\cdot H^T = 0$, osea que H genera a un código $\mathcal{C}'$ que es ortogonal a $\mathcal{C}$ ($\mathcal{C}' \subset C^{\perp}$). Como además se verifica que $h_{n-r} \neq 0$, entonces $rg(H) = n-r$ y $dim(\mathcal{C}') = n-r$ y $\mathcal{C}' = \mathcal{C}^{\perp}$.

3. Por último, si $\mathcal{C} = \ll g(x) \gg$, se tiene que $h(x)\cdot g(x) = x^n - 1$, y entonces $h(x) = \frac{x^n-1}{g(x)}$. Ahora bien,

$$\begin{array}{c} h(x) = h_0 + h_1\,x + ... + h_{n-r}\,x^{n-r} \Rightarrow \\ \Rightarrow h(x^{-1}) = h_0 + h_1\,x^{-1} + ... + h_{n-r}\,x^{r-n} \Rightarrow \\ \Rightarrow x^{n-r}\,h(x^{-1}) = h_0\,x^{n-r} + h_1\,x^{n-r-1} + ... + h_{n-r} \Rightarrow \\ \Rightarrow \frac{1}{h_0}\,x^{n-r}\,h(x^{-1}) = h_0^{-1}\,x^{n-r}\,h(x^{-1}) = g^{\perp}(x) \end{array}$$

Donde se ve que $g^{\perp}(x)$ divide a $x^n - 1$ (ya que $h(x^{-1})\cdot g(x^{-1}) = x^{-n} - 1$, luego $x^{n-r}\cdot h(x^{-1})\cdot x^r\cdot g(x^{-1}) = 1 - x^n$). Por lo tanto, $g^{\perp}(x)$ genera un código cíclico.

Sea ahora $\mathcal{C}' = \ll g^{\perp}(x) \gg$, que tiene como matriz generadora a H y, a su vez, H es la matriz de control de $\mathcal{C}$, por consiguiente, $\mathcal{C}' = \mathcal{C}^{\perp} = \ll g^{\perp}(x) \gg$, y $\mathcal{C}^{\perp}$ es cíclico. □

Ejemplo 2.2.1. *Volviendo al ejemplo 2.1.7 donde escribimos completamente el código $\mathcal{C} = \ll x^3+x+1 \gg$, veamos ahora quién sería la matriz generadora y la matriz de paridad de dicho código:*

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \in M_{4\times 7}(\mathbb{F}_2)$$

Y como $x^7 - 1 = (x^3 + x + 1)(x^4 + x^2 + x + 1)$, entonces la matriz de control es:

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \in M_{3\times 7}(\mathbb{F}_2)$$

2.3. Ceros de Polinomios

En este apartado, vamos a ver una forma alternativa de describir un código cíclico de longitud n a través de las ceros del polinomio generador, es decir, por medio de ciertas raíces n-ésimas de la unidad.

Definición 2.3.1. *Si $f(x) \in \mathbb{F}_q[x]$, el **cuerpo de descomposición** de $f(x)$ sobre $\mathbb{F}_q$ es el menor cuerpo que contiene a $\mathbb{F}_q$ y a todas las raíces de $f(x)$. Se denota por $SF(f(x))_{\mathbb{F}_q}$.*

En particular, si consideramos el polinomio $x^n - 1$ se tiene que:

$$SF(x^n - 1)_{\mathbb{F}_q} = \mathbb{F}_{q^{o_n(q)}} = \mathbb{F}_q(\alpha)$$

donde α es cualquier raíz primitiva de $x^n - 1$ y $o_n(q)$ es el orden de q módulo n (es decir, el menor entero positivo s tal que $q^s \equiv 1(mod\, n)$).

Ahora, sea

$$x^n - 1 = \prod_i m_i(x)$$

la factorización de $x^n - 1$ en irreducibles mónicos sobre $\mathbb{F}_q$ y sea α una raíz de $m_i(x)$ en alguna extensión de cuerpo $\mathbb{F}_{q^d}$ de $\mathbb{F}_q$ (osea, $\mathbb{F}_{q^d} \subset \mathbb{F}_{q^{o_n(q)}}$, con $d|o_n(q)$). De esta forma, $m_i(x) = m_\alpha(x)$ es el polinomio mínimo de α sobre $\mathbb{F}_q$ (polinomio mónico de menor grado para el cual α es raíz). Así, si $f(x) \in A_{q,n}$, entonces $f(\alpha) = 0 \Leftrightarrow f(x) \in \ll m_i(x) \gg$.

Como generalización encontramos el siguiente teorema:

Teorema 2.3.1. *Sea $g(x) = q_1(x) \cdot q_2(x) ... q_t(x)$ un producto de factores irreducibles de $x^n - 1$ sobre $\mathbb{F}_q$, y sean $\{\alpha_1 ... \alpha_r\}$ las raíces de $g(x)$ en $SF(x^n - 1)_{\mathbb{F}_q}$. Entonces:*

$$\mathcal{C} = \ll g(x) \gg = \{f(x) \in A_{q,n} : f(\alpha_1) = 0 ... f(\alpha_r) = 0\}$$

De hecho, basta tomar una raíz de cada factor irreducible de $g(x)$. Esto es, si β_i es una raíz de q_i para $1 \leq i \leq t \leq r$, entonces:

$$\mathcal{C} = \ll g(x) \gg = \{f(x) \in A_{q,n} : f(\beta_1) = 0, ..., f(\beta_t) = 0\}$$

Demostración. Para esta demostración, sea $\mathcal{C} = \ll g(x) \gg$ y tomamos los conjuntos:

$$\mathcal{S}_{\alpha_1,...,\alpha_r} = \{f(x) \in A_{q,n} : f(\alpha_1) = 0, ..., f(\alpha_r) = 0)\}$$

$$\mathcal{S}_{\beta_1,...,\beta_t} = \{f(x) \in A_{q,n} : f(\beta_1) = 0, ..., f(\beta_r) = 0)\}$$

Se observa que $\mathcal{S}_{\alpha_1,...,\alpha_r} \subseteq \mathcal{S}_{\beta_1,...,\beta_t}$, y vamos a ver que $\mathcal{C} = \mathcal{S}_{\alpha_1 ... \alpha_r} = \mathcal{S}_{\beta_1 ... \beta_t}$. Lo haremos por doble inclusión:

$\boxed{\subseteq}$ Sea $f(x) \in \mathcal{C}$. Lógicamente, $f(x) = a(x) \cdot g(x)$; entonces, $f(\alpha_i) = 0$, para $1 \leq i \leq r$. Esto significa que $f(x) \in \mathcal{S}_{\alpha_1,...,\alpha_r}$, lo que nos lleva a que $\mathcal{C} \subseteq \mathcal{S}_{\alpha_1,...,\alpha_r}$.

$\boxed{\supseteq}$ Supongamos ahora que $f(x) \in \mathcal{S}_{\beta_1,...,\beta_t}$. Esto implica que $f(\beta_i) = 0$ y como además $q_i(x)$ es irreducible y mónico y $q_i(\beta_i) = 0$, entonces $q_i(x)|f(x)$ para $1 \leq i \leq t$; y esto conlleva que:

$$f(x) \in \bigcap_{i=1}^{t} \ll q_i(x) \gg = \ll m.c.m\{q_1(x)...q_t(x)\} \gg$$

A continuación, como q_i es irreducible sobre $\mathbb{F}_q$ y no tiene raíces comunes con q_j, se tiene que:

$$f(x) \in \ll m.c.m\{q_1(x)...q_t(x)\} \gg = \ll \prod_{i=1}^{t} q_i(x) \gg = \ll g(x) \gg$$

Y de aquí se concluye que $\mathcal{S}_{\beta_1...\beta_t} \subseteq \mathcal{C}$; y finalmente $\mathcal{C} = \mathcal{S}_{\alpha_1...\alpha_r} = \mathcal{S}_{\beta_1...\beta_t}$.

□

Observación 2.3.1. *Si $\alpha_1...\alpha_r$ es un conjunto de raíces de $x^n - 1$, entonces el código $\mathcal{C} = \mathcal{S}_{\alpha_1...\alpha_r}$ tiene como polinomio generador a:*

$$g(x) = m.c.m\{m_{\alpha_1}(x)...m_{\alpha_r}(x)\}$$

Las raíces del polinomio generador de un código cíclico C se denominan **ceros de** $\mathcal{C}$. Así, la descripción de códigos cíclicos a través de sus ceros permiten definir de forma sencilla familias conocidas de códigos cíclicos, como los códigos BCH o los códigos Reed-Solomon.

Veamos ahora un ejemplo de cómo construir una matriz de paridad a partir de los ceros de un código $\mathcal{C}$.

Ejemplo 2.3.1. *Consideremos la extensión $\mathbb{F}_{2^3} \cong \frac{\mathbb{F}_2[x]}{(x^3+x+1)}$. Supongamos $n = 7$ y que tenemos el siguiente conjunto de ceros:*

$$\mathcal{Z} = \{1, \alpha, \alpha^2, \alpha^4\}$$

En este caso, tendríamos el código $\mathcal{C} = \ll x^4 + x^3 + x^2 + x + 1 \gg$.

Además, $c(x) = c_0 + ... + c_6\, x^6 \in \mathcal{C}$ si, y sólo si, $c(\alpha^j) = 0$ para $j \in \{0, 1, 2, 4\}$, y esto ocurre si, y sólo si, $c(1) = c(\alpha) = 0$. Y en forma matricial, se tiene que:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} = 0$$

Ahora, como $\mathbb{F}_{2^3} \cong \mathbb{F}_2^3$, si consideramos los términos de la matriz anterior en $\mathbb{F}_2^3$ obtenemos la matriz:

$$\widetilde{H} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \in M_{6\times 7}(\mathbb{F}_2)$$

donde se observa que las filas son linealmente independientes, es decir que $rg(\widetilde{H}) = 6$ *y como ya vimos,* $\widetilde{H}\, c^T = 0$*, podemos afirmar que* $H = \widetilde{H}$ *es una matriz de control de este código.*

2.4. Codificación cíclica

Si $\mathcal{C} = \ll g(x) \gg$ es un código cíclico q-ario de longitud n, y $deg(g(x)) = r$, es decir, si $\mathcal{C}$ es un $[n,k]_q$-código con $k = n - r$, para codificar una palabra de $\mathbb{F}_q^k$ se puede utilizar alguno de los métodos siguientes:

- **Método 1: Codificación no sistemática**

  Este método consiste en usar la matriz generadora a partir del polinomio generador y de la base $\{g(x), x\cdot g(x), ..., x^{k-1}\cdot g(x)\}$ que ya habíamos visto. Supongamos que tenemos la palabra $a = (a_0, ..., a_{k-1}) \in \mathbb{F}_q^k$, entonces la codificación consiste simplemente en el producto $(a_0, ..., a_{k-1})\cdot G \in C \subset \mathbb{F}_q^n$. Esto expresado polinómicamente sería:

  $$a_0\cdot g(x) + a_1\cdot x\cdot g(x) + ... + a_{k-1}\cdot x^{k-1}\cdot g(x) =$$

  $$(a_0 + a_1\cdot x + ... + a_{k-1}\cdot x^{k-1})\cdot g(x) = a(x)\cdot g(x) \in \mathcal{C} = \ll g(x) \gg$$

  .

- **Método 2: Codificación sistemática**

  Recordemos que habíamos encontrado que la matriz:

  $$G = \left(\begin{array}{c|c} \begin{matrix} -r_{n-k}(x) \\ -r_{n-k+1}(x) \\ \vdots \\ -r_{n-1}(x) \end{matrix} & I_k \end{array}\right)$$

  proporcionaba una codificación sistemática donde las $n-r$ últimas coordenadas son símbolos de información; mientras que las restantes son redundantes.

  $$(a_0, ..., a_{k-1})\cdot G = (a_k, ..., a_{n-1}, a_0, ..., a_{k-1})$$

siendo:

$$\begin{aligned} a_k &= -(a_0 \cdot r_{n-k,0} + a_1 \cdot r_{n-k+1,0} + ... + a_{k-1} \cdot r_{n-1,0}) \\ &\vdots \\ a_{n-1} &= -(a_0 \cdot r_{n-k,n-k-1} + a_1 \cdot r_{n-k+1,n-k-1} + ... + a_{k-1} \cdot r_{n-1,n-k-1}) \end{aligned}$$

Expresado en forma polinomial tendríamos:

$$-[\sum_{i=0}^{k-1} a_i\, r_{n-k+i,0} + ... + x^{n-k-1} \cdot \sum_{i=0}^{k-1} a_i\, r_{n-k+i,n-k-1}] + \sum_{i=0}^{k-1} a_i\, x^{n-k+i} =$$

$$-[\sum_{i=0}^{k-1} a_i\, r_{n-k+i}(x)] + \sum_{i=0}^{k-1} a_i\, x^{n-k+i} = x^{n-k} \cdot a(x) - [\sum_{i=0}^{k-1} a_i\, r_{n-k+i}(x)]$$

$$x^{n-k} \cdot a(x) - \sum_{i=0}^{k-1} a_i\, (x^{n-k+i} - g(x) \cdot q_{n-k+i}(x)) =$$

$$= x^{n-k} \cdot a(x) - x^{n-k}(a_0 + ... + a_{k-1} \cdot x^{k-1}) - g(x) \cdot q(x)$$

De aquí se deduce que:

$$x^{n-k}(a_0 + ... + a_{k-1} \cdot x^{k-1}) = g(x) \cdot q(x) + (a_0 \cdot r_{n-k}(x) + ... + a_{k-1} \cdot r_{n-1}(x))$$

con $deg(a_0 \cdot r_{n-k}(x) + ... + a_{k-1} \cdot r_{n-1}(x)) \leq n - k < r$. Luego,

$$a_0 \cdot r_{n-k}(x) + ... + a_{k-1} \cdot r_{n-1}(x) = R(x^{n-k} \cdot a(x), g(x))$$

Con todo esto, podemos concluir que el segundo método sería:

1. Calcular $x^{n-k} \cdot a(x)$.
2. Dividir $x^{n-k} \cdot a(x)$ y $g(x)$ con el fin de obtener $R(x^{n-k} \cdot a(x), g(x))$.
3. Calcular $x^{n-k} \cdot a(x) - R(x^{n-k} \cdot a(x), g(x)) = c(x) \in C$.

- **Método 3: Codificación a través de la matriz de paridad**

  Sea H la matriz de paridad del código. Sabemos que tiene rango $n - k = r$:

$$c = (c_0 ... c_{n-1}) \in C \Leftrightarrow c \cdot H^T = 0$$

y esto sucede si y sólo si c satisface las ecuaciones de paridad:

$$\left.\begin{aligned} c_0 \cdot h_k + c_1 \cdot h_{k-1} + ... + c_k \cdot h_0 &= 0 \\ c_1 \cdot h_k + c_2 \cdot h_{k-1} + ... + c_{k+1} \cdot h_0 &= 0 \\ &\vdots \\ c_{r-1} \cdot h_k + c_r \cdot h_{k-1} + ... + c_{k+r-1} \cdot h_0 &= 0 \end{aligned}\right\}$$

De esta manera, si conocemos las primeras k coordenadas $c_0...c_{k-1}$, podemos calcular las coordenadas restantes $c_k...c_{n-1}$ despejando del sistema anterior.

Es fácil ver que este método de codificación es sistemático.

Ejemplo 2.4.1. *Usemos los tres métodos antes explicados para un* $[7,3,4]$*-código cíclico* $\mathcal{C}$ *generado por* $g(x) = x^4 + x^2 + x + 1$.

Vamos a considerar el mensaje $a = (011)$*, es decir,* $a(x) = x + x^2$.

1. *Haciendo uso del primer método, simplemente tenemos que:*

$$c(x) = a(x)\cdot g(x) = (x + x^2)(x^4 + x^2 + x + 1) = x^6 + x^5 + x^4 + x = (0100111)$$

Esto significa que:

$$a = (011) \longmapsto c = (0100111) \in \mathcal{C}$$

2. *Sigamos ahora los pasos del segundo método:*

a) *Calculamos* $x^{n-k}\cdot a(x) = x^4(x + x^2) = x^5 + x^6$.

b) *Dividimos* $x^5 + x^6$ *y* $x^4 + x^2 + x + 1$*, de modo que:*

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

Por lo que $R(x^5 + x^6, x^4 + x^2 + x + 1) = x^2 + 1$

c) *Tenemos que* $c(x) = x^6 + x^5 + x^2 + 1$*. Es decir que:*

$$a = (011) \longmapsto c = (1010011) \in \mathcal{C}$$

y vemos que el mensaje se recupera en las últimas coordenadas.

3. *En este último método, tomamos* $c = 011c_3c_4c_5c_6$*, donde* c_3*,* c_4*,* c_5 *y* c_6 *deben satisfacer las ecuaciones:*

$$\left.\begin{array}{l} c_0\cdot h_3 + c_1\cdot h_2 + c_2\cdot h_1 + c_3\cdot h_0 = 0 \\ c_1\cdot h_3 + c_2\cdot h_2 + c_3\cdot h_1 + c_4\cdot h_0 = 0 \\ c_2\cdot h_3 + c_3\cdot h_2 + c_4\cdot h_1 + c_5\cdot h_0 = 0 \\ c_3\cdot h_3 + c_4\cdot h_2 + c_5\cdot h_1 + c_6\cdot h_0 = 0 \end{array}\right\}$$

Pero en nuestro caso, $c_0 = 0$ *y* $c_1 = c_2 = 1$*, y además,* $h(x) = \frac{x^7-1}{x^4+x^2+x+1} = x^3+x+1$ *así que* $h_2 = 0$ *y* $h_0 = h_1 = h_3 = 1$*. Luego, el sistema anterior queda reducido a:*

$$\left.\begin{array}{r} 1 + c_3 = 0 \\ 1 + c_3 + c_4 = 0 \\ 1 + c_4 + c_5 = 0 \\ c_3 + c_5 + c_6 = 0 \end{array}\right\} \Rightarrow \left.\begin{array}{l} c_3 = 1 \\ c_4 = 0 \\ c_5 = 1 \\ c_6 = 0 \end{array}\right\}$$

Por tanto, finalmente la codificación en este caso será:

$$a = (011) \longmapsto c = (0111010) \in \mathcal{C}$$

Se observa que cada método aporta una codificación distinta, pese a ser el mismo código.

2.5. Detección y corrección de errores. Decodificación de códigos cíclicos

Un código $\mathcal{C}$ detecta s errores si cuando en una palabra se comete un número menor de errores, la palabra resultante no es del código $\mathcal{C}$. Asimismo, se dice que el código es **s-detector** si detecta s errores pero no $s+1$.

Análogamente, un código corrige t errores si al decodificar por la mínima distancia, se pueden corregir todos los errores de peso t o menor. En este caso, se dice que el código es **t-corrector** si corrige t errores pero no $t+1$.

Siempre va a ocurrir que si $C \subseteq A_{q,n}$ es un código cuya distancia mínima es d, entonces $\mathcal{C}$ es $(d-1)$-detector y $\lfloor \frac{d-1}{2} \rfloor$-corrector.(

Para la decodificación cíclica, usaremos el conocido como **algoritmo del líder** o **decodificación por síndrome**, aunque de una forma más reducida que la habitual para códigos lineales.

Supondremos que si $c(x) \in \mathcal{C}$ es el código enviado y $u(x)$ el polinomio recibido, entonces $e(x) = u(x) - c(x)$ es el **polinomio error**.

Definición 2.5.1. *Sea $a(x) = a_0 + ... + a_{n-1} \cdot x^{n-1} \in A_{q,n}$, entonces el **síndrome** se define como;*

$$s(a(x)) = (a_0 ... a_{n-1}) \cdot H^T$$

Si pensamos en el código $\mathcal{C} = \ll g(x) \gg$, habíamos visto que una matriz generadora estándar es:

$$G = \left(\begin{array}{c|c} -r_{n-k}(x) & \\ -r_{n-k+1}(x) & \\ \vdots & I_k \\ -r_{n-1}(x) & \end{array} \right)$$

por lo que una matriz de control es:

$$H = \left(\begin{array}{c|ccc} I_{n-k} & r_{n-k}(x) & ... & r_{n-1}(x) \end{array} \right)$$

lo que nos lleva a que:

$$s(a(x)) = (a_0 ... a_{n-1}) \cdot H^T = (a_0 + ... + a_{n-k-1} \cdot x^{n-k-1}) + (a_{n-k} \cdot r_{n-k}(x) + ... + a_{n-1} \cdot r_{n-1}(x))$$

Y así:

$$s(a(x)) - a(x) = \sum_{i=0}^{n-k-1} a_i \, x^i + \sum_{i=n-k}^{n-1} a_i \cdot r_i(x) - \sum_{i=0}^{n-1} a_i \, x^i = \sum_{i=n-k}^{n-1} a_i \cdot r_i(x) - \sum_{i=n-k}^{n-1} a_i \, x^i =$$

$$\sum_{i=n-k}^{n-1} a_i (r_i(x) - x^i) = - \sum_{i=n-k}^{n-1} a_i \cdot g(x) \cdot q_i(x) = -g(x) \cdot q(x)$$

Por lo tanto, $a(x) = g(x) \cdot q(x) + s(a(x))$ y $deg(s(a(x))) \leq n-k-1 < n-k = r = deg(g(x))$ o $deg(s(a(x))) = 0$: lo cual significa que:

$$s(a(x)) = R(a(x), g(x))$$

En particular, si recibimos un mensaje $a(x) \in A_{q,n}$, está claro que:

$$a(x) \in C = \ll g(x) \gg \Longleftrightarrow R(a(x), g(x)) = 0 \Longleftrightarrow s(a(x)) = 0$$

También cabe destacar que $s(a_1(x)) = s(a_2(x)) \Leftrightarrow a_1(x)$ y $a_2(x)$ están en la misma coclase determinada por $\mathcal{C}$ en $A_{q,n}$ (esto se ve inmediato al haberse llegado a que $s(a(x)) = R(a(x), g(x))$).

En definitiva, supongamos que nos llega el polinomio $u(x) = c(x) + e(x)$ tal que el error tiene peso $w(e(x)) \leq t = \lfloor \frac{d-1}{2} \rfloor$. Entonces, si podemos determinar $e(x)$, decodificamos $u(x)$ como $c(x) = u(x) - e(x)$.

Este motivo nos lleva a intentar encontrar $e(x)$. Esta búsqueda se basa en que $s(u(x)) = s(e(x))$ (ya que la linealidad del síndrome nos permite hacer $s(u(x)) = s(a(x) + e(x)) = s(a(x)) + s(e(x)) = s(e(x))$). Así pues, calculamos $s(u(x))$ y tomamos el polinomio error $e(x)$ como el polinomio de menor peso en la coclase de $u(x)$, es decir, con sídrome $s(u(x))$. A este polinomio que tomamos se le denomina **líder de la coclase**.

Ejemplo 2.5.1. *En este ejemplo, recurrimos al código* $\mathcal{C} = \ll x^3 + x^2 + 1 \gg$ *en* $\mathbb{F}_2[x]$.

Ocurre que en este código $d = 3$*, lo cual quiere decir que* $\mathcal{C}$ *corrige los errores de peso* 1*. Así, la tabla de líderes de las coclases y sus síndromes queda recogida a continuación:*

$e(x)$	$s(e(x))$
0	0
1	1
x	x
x^2	x^2
x^3	$x^2 + 1$
x^4	$x^2 + x + 1$
x^5	$x + 1$
x^6	$x^2 + x$

Ahora, supongamos que recibimos el polinomio $u(x) = x^5 + 1$*. Hallamos su síndrome:*

$$s(u(x)) = R(u(x), g(x)) = R(x^5 + 1, x^3 + x^2 + 1) = x$$

ya que $x^5 + 1 = (x^2 + x + 1)(x^3 + x^2 + 1) + x$*. Conociendo el síndrome de la palabra recibida ya sólo queda ir a la tabla para ver que* $e(x) = x$ *y , por consiguiente,* $c(x) = u(x) - e(x) = x^5 + x + 1$*.*

O sea que hemos recibido $u = (1000010)$ *con al menos un error (pues* $u \notin \mathcal{C}$*) y, asumiendo un único error, lo decodificamos como* $c = (1100010) \in \mathcal{C}$*.*

Observación 2.5.1. *El algoritmo de decodificación que acabamos de ver se puede mejorar usando el hecho de que* $\mathcal{C}$ *sea un código cíclico. Para ello, hemos de suponer que tenemos algún método para decodificar el* ***coeficiente principal*** *de cualquier palabra recibida:*

$$u(x) = u_{n-1} \cdot x^{n-1} + ... + u_1 \cdot x + u_0$$

que podemos suponer que es u_{n-1}*. Así, el proceso consiste en:*

1. *Decodificar* u_{n-1}*.*
2. *Hacer* $x \cdot u(x)$ *de forma que el nuevo coeficiente principal es* u_{n-2}*.*
3. *Decodificar* u_{n-2}*.*
4. *Repetir este procedimiento hasta decodificar toda la palabra.*

Con este método, sólo necesitamos las filas de la tabla de líderes y síndromes que contienen líderes de grado $n-1$*. Apliquémoslo al ejemplo anterior:*

Ejemplo 2.5.2. *Recordemos que queremos decodificar* $u(x) = x^5 + 1$*. Como sólo hay un líder de peso* 1 *y grado* 6*, nos reducimos a la tabla:*

$e(x)$	$s(e(x))$
x^6	$x^2 + x$

Ahora, como $s(u(x)) = x$ *que no está en la tabla entonces damos por supuesto que* x^6 *es correcto. Continuamos con* $x \cdot u(x) = x^6 + x$ *cuyo síndrome es* x^2 *por lo que también la damos por correcta. En el caso de* $x^2 \cdot u(x) = (x^2 + 1)$ *tenemos que* $s(x^2 \cdot u(x)) = x^2 + 1$ *que tampoco está en la tabla; así que el coeficiente de* x^4 *en* $u(x)$ *es correcto. Proseguimos con* $x^3 \cdot u(x) = x^3 + x$*, cuyo sídrome es* $s(x^3 \cdot u(x)) = x^2 + x + 1$*, que tampoco está en la tabla por lo que el coeficiente de* x^3 *sí que es correcto. Siguiendo de esta forma,* $x^4 \cdot u(x) = x^4 + x^2$ *y* $s(x^4 \cdot u(x)) = x + 1$ *motivo por el cual el coeficiente de* x^2 *está bien. No sucede lo mismo si hacemos* $x^5 \cdot u(x) = x^5 + x^3$*, puesto que* $s(x^5 + x^3) = x^2 + x$ *que sí está en la tabla; luego el coeficiente que acompaña a* x *es erróneo. Por último, veríamos que* $s(x^6 \cdot u(x)) = s(x^6 + x^4) = 1$*, o sea que el témino independiente es correcto y no hay más errores. Así, decodificaríamos la palabra como* $c(x) = u(x) - x = x^5 + x + 1$ *que es justamente lo que decodificábamos con el otro método.*

Capítulo 3

Códigos BCH y Códigos Reed-Solomon

3.1. Definición de los códigos BCH, ejemplos y propiedades

Ya hemos visto que un código cíclico puede definirse a través de ceros de polinomios. En particular, si $\alpha_1 \dots \alpha_n$ son raíces n-ésimas de la unidad en $\mathbb{F}_{q^t}$, entonces el código:

$$\mathcal{C} = \{p(x) \in A_{q,n} : p(\alpha_1) = 0, \dots ,p(\alpha_n) = 0\}$$

es un código cíclico cuyo polinomio generador $g(x)$ es el producto de los polinomios mínimos de $\alpha_1 \dots \alpha_n$ en $\mathbb{F}_q$. Esto da idea de que se pueden encontrar algunos códigos interesantes especificando ciertos conjuntos especiales de raíces de la unidad como ceros de $g(x)$. Así surgen los Códigos BCH.

Estos códigos fueron descubiertos por R.C.Bose, D.K.Ray-Chauduri y R.C.Hocquenhem en 1960, y de ahí su nombre. Su interés y principal característica reside en la facilidad para codificarlos y decodificarlos.

Teorema 3.1.1. *(**COTA BCH**) Sea ω una raíz n-ésima primitiva de la unidad en $\mathbb{F}_{q^t}$. Sea además $\mathcal{C}$ un código cíclico en $A_{q,n}$ cuyo polinomio generador $g(x)$ es el polinomio mónico de menor grado en $\mathbb{F}_q[x]$ que tiene a los $\delta - 1$ números:*

$$\omega^b, \omega^{b+1}, \dots , \omega^{b+\delta-1}$$

entre sus ceros, donde $b \geq 0$ y $\delta \geq 1$. Entonces $\mathcal{C}$ tiene distancia mínima al menos δ. Es decir:

$$d(\mathcal{C}) = w(\mathcal{C}) \geq \delta + 1$$

Demostración. [7], Corolario 7.5.4 página 343. □

Este teorema en realidad nos está diciendo que es posible obtener un código con distancia mínima δ, especificando que su polinomio generador tenga $\delta - 1$ ceros con exponentes consecutivos. Y esto motiva la siguiente definición:

Definición 3.1.1. *Sea ω una raíz n-ésima primitiva de la unidad en $\mathbb{F}_{q^t}$. Un* ***Código BCH con distancia diseñada*** *δ es un código cíclico en $A_{q,n}$ (q-ario y de longitud n) cuyo polinomio generador $g(x)$ es el polinomio mónico de menor grado en $\mathbb{F}_q[x]$ cuyos ceros son los $\delta - 1$ números:*

$$\omega^b, \omega^{b+1}, \ \dots \ , \omega^{b+\delta-2}$$

donde $b \geq 0$ y $\delta \geq 1$. Luego,

$$g(x) = m.c.m\{m_b(x), m_{b+1}(x),..., m_{b+\delta-2}(x)\}$$

Se denota por $\mathcal{B}_q(n, \delta, \omega, b)$.

De esta forma, algunos códigos BCH usuales son:

- Si $b = 1$, entonces tenemos un *Código BCH en sentido estricto o estrecho*. Y, en este caso, se denota simplemente por $\mathcal{B}_q(n, \delta, \omega)$.
- Si la longitud n es de la forma $n = q^m - 1$, con $n \in \mathbb{N}$, entonces se dice que $\mathcal{C}$ es un *Código BCH primitivo*. En esta ocasión, α es un elemento primitivo de $\mathbb{F}_{q^m}$ y m coincide con el orden de q módulo n.
- Si $m = 1$, entonces obtenemos el conocido *Código Reed-Solomon*, que se estudiará más detalladamente en la siguiente sección.

Cabe aclarar que de la definición de Código BCH se puede ver que:

$$\mathcal{B}_q(n, \delta, \omega, b) = \{p(x) \in A_{q,n} : p(\omega^b) = p(\omega^{b+1}) =, \ \dots \ ,= p(\omega^{b+\delta-2}) = 0\}$$

De la propia definición de los códigos BCH, se observa que si $\delta_1 < \delta_2$ entonces el código $\mathcal{B}_q(n, \delta_2, \omega, b)$ está contenido en $\mathcal{B}_q(n, \delta_1, \omega, b)$, y siguiendo el mismo razonamiento sucesivamente, ocurre que:

$$\mathcal{B}_q(n, \delta, \omega, b) \subseteq \mathcal{B}_q(n, \delta - 1, \omega, b) \subseteq ... \subseteq \mathcal{B}_q(n, 1, \omega, b)$$

Por otra parte, ya se ha visto que la dimensión de un código cíclico viene dada por $k = n - deg(g(x))$. Y, como también $deg(m_{b+i}(x)) \leq o_n(q)$, entonces podemos afirmar que:

$$dim(\mathcal{B}_q(n, \delta, \omega, b)) = k = n - deg(g(x)) \geq n - o_n(q) \cdot (\delta - 1)$$

Observación 3.1.1. *Según el teorema 3.1.1, para un código BCH se cumple que la distancia $d(\mathcal{B}_q(n, \delta, \omega, b)) \geq \delta$; pero, en general, es complicado calcular la distancia mínima de un código BCH, por lo que se suele tomar δ como sustituto de la distancia real.*

Con el fin de hacer un análisis más a fondo de los códigos BCH en el sentido estricto ($b = 1$), observamos que en el caso binario, se verifican las siguientes equivalencias:

$$m_i(\omega) = 0 \Longleftrightarrow [m_i(\omega)]^2 = 0 \Longleftrightarrow m_i(\omega^2) = 0$$

Pero esto implica a su vez que, para $m \geq 1$, los tres polinomios:

- $g_1(x) = m.c.m\{m_1(x), m_2(x), ..., m_{2m}(x)\}$
- $g_2(x) = m.c.m\{m_1(x), m_2(x), ..., m_{2m-1}(x)\}$
- $g_3(x) = m.c.m\{m_1(x), m_3(x), ..., m_{2m-1}(x)\}$

son idénticos. En particular, como $g_2(x) = g_1(x)$, tenemos que:

$$\mathcal{B}_2(n, 2m+1, \omega) = \mathcal{B}_2(n, 2m, \omega)$$

De donde se concluye sin problema que basta restringirse al estudio de los códigos BCH binarios con distancia δ impar.

Por otra parte, el hecho de que $g_3(x) = g_1(x)$ permite mejorar la cota $dim(\mathcal{B}_2(n, \delta, \omega)) \geq n - o_n(2)\cdot(\delta - 1)$ de la siguiente forma:

$$dim(\mathcal{B}_2(n, 2m+1, \omega)) \geq n - m\, o_n(2)$$

Ejemplo 3.1.1. *En este ejemplo se hallan los códigos BCH binarios en el sentido estricto de longitud 7, esto es $\mathcal{B}_2(7, \delta, \omega)$. Por ello, sea $n = 7$, $q = 2$ y $b = 1$.Los conjuntos ciclotómicos módulo 2 son:*

$$\begin{aligned} C_0 &= \{0\} \\ C_1 &= \{1, 2, 4\} \\ C_3 &= \{3, 5, 6\} \end{aligned}$$

Y por consiguiente, se tiene que:

- $m_0(x) = x - 1$
- $m_1(x) = m_2(x) = m_4(x) = (x-\omega)(x-\omega^2)(x-\omega^4) =$
  $= x^3 + (\omega^4 + \omega^2 + \omega)\cdot x^2 + (\omega^6 + \omega^5 + \omega^3)\cdot x + 1$
  $= x^3 + x + 1$
- $m_3(x) = m_5(x) = m_6(x) = (x-\omega^3)(x-\omega^6)(x-\omega^6) =$
  $= x^3 + (\omega^6 + \omega^5 + \omega^3)\cdot x^2 + (\omega^4 + \omega^2 + \omega)\cdot x + 1 =$
  $= x^3 + x^2 + 1$

Lo cual genera los siguientes códigos en función de los valores de la distancia diseñada δ:

- *Si $\delta = 1$.*
  No hay ceros, así que el código es de la forma $\mathcal{B}_2(7, 1, \omega) = \ll 1 \gg = \mathbb{F}_2^7$.
- *Si $\delta = 3$.*
  Los ceros del polinomio generador son ω y ω^2, pues $b + \delta - 2 = 1 + 3 - 2 = 2$; y, por tanto, $g(x) = m.c.m\{m_1(x), m_2(x)\} = m_1(x) = x^3 + x + 1$. De esta forma el código en este caso es $\mathcal{B}_2(7, 3, \omega) = \ll x^3 + x + 1 \gg$.

- *Si $\delta = 5$.*
  En esta ocasión, las raíces del polinomio generador son ω, ω^2, ω^3 y ω^4, puesto que $b + \delta - 2 = 1 + 5 - 2 = 4$; luego,

$$g(x) = m.c.m\{m_1(x), m_2(x), m_3(x), m_4(x)\} = m.c.m\{m_1(x), m_3(x)\} = \\ = m.c.m\{x^3 + x + 1, x^3 + x^2 + 1\} = (x^3 + x + 1)(x^3 + x^2 + 1) = \\ = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

  Y el código es $\mathcal{B}_2(7, 5, \omega) = \ll x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \gg$.

- *Si $\delta = 7$.*
  Esta vez los ceros del polinomio generador son ω, ω^2, ω^3, ω^4, ω^5 y ω^6, dado que $b + \delta - 2 = 1 + 7 - 2 = 6$; así que,

$$g(x) = m.c.m\{m_1(x), m_2(x), m_3(x), m_4(x), m_5(x), m_6(x)\} = \\ = m.c.m\{m_1(x), m_3(x)\} = m.c.m\{x^3 + x + 1, x^3 + x^2 + 1\} = \\ = (x^3 + x + 1)(x^3 + x^2 + 1) = \\ = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1.$$

  Lo que conduce al código $\mathcal{B}_2(7, 7, \omega) = \ll x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \gg$.

Estos casos quedan resumidos en la tabla que sigue estas líneas:

Distancia diseñada	**Polinomio generador**	**Dimensión**	**Distancia**
1	1	7	1
3	$m_1(x)$	4	3
5	$m_{1,3}(x)$	1	7
7	$m_{1,3}(x)$	1	7

Teorema 3.1.2. *Si $\delta | n$, entonces el código BCH binario en el sentido estricto $\mathcal{B}_2(n, \delta, \omega)$ tiene distancia mínima $d = \delta$*

Demostración. Sabemos que:

$$\delta | n \Rightarrow \exists k \in \mathbb{Z} : n = k \cdot \delta \Rightarrow x^n - 1 = x^{k \cdot \delta} - 1 = \\ = (x^k)^\delta - 1 = (x^k - 1)(x^{(\delta-1)\cdot k} + ... + x^{2k} + x^k + 1)$$

Y como ω es una raíz primitiva, entonces $\omega^{ik} \neq 1$ para $i = 1, 2, ..., \delta - 1$. Y esto implica que $\omega, \omega^2, ..., \omega^{\delta-1}$ no son ceros de $x^k - 1$; así que tienen que ser ceros de:

$$p(x) = x^{(\delta-1)\cdot k} + ... + x^{2k} + x^k + 1$$

Ahora bien, $p(x) \in \mathcal{B}_2(n, \delta, \omega)$ y tiene peso δ. Concluyendo que, como en un código cíclico la distancia coincide con el peso, entonces $d = \delta$. □

Para continuar, pasemos a considerar los códigos BCH de la forma:

$$\mathcal{B} = \mathcal{B}_2(2^s - 1, \delta, \omega)$$

con $s \in \mathbb{N}$, δ impar y ω raíz n-ésima primitiva de la unidad.

Se puede comprobar que la distancia mínima $d_{\mathcal{B}}$ de este tipo de códigos es impar. [7] (pág 358-360). Y así, el siguiente teorema cobra sentido:

Teorema 3.1.3. *Sea $\mathcal{B} = \mathcal{B}_2(2^s - 1, 2m + 1, \omega)$. Entonces,*

$$2^{m \cdot s} < \sum_{i=0}^{m+1} \binom{n}{i} \Rightarrow d_{\mathcal{B}} = \delta$$

Demostración. Ya hemos dicho que los códigos BCH verifican que $d_{\mathcal{B}} \geq \delta$. Además, también hemos comentado que es posible comprobar que $d_{\mathcal{B}}$ es impar. Supongamos que $d_{\mathcal{B}} > \delta$.

$$d_{\mathcal{B}} > \delta \Rightarrow d_{\mathcal{B}} > 2m + 1 \Rightarrow d_{\mathcal{B}} \geq 2m + 3 = 2(m + 1) + 1$$

Por otra parte, ya hemos visto que:

$$k = dim(\mathcal{B}) \geq n - m \cdot o_n(2) = n - m \cdot s$$

Ahora, usando la cota de Hamming:

$$|c| \cdot \sum_{i=0}^{m+1} \binom{n}{i} \leq 2^{m \cdot s}$$

se tiene que:

$$2^{n-m \cdot s} \cdot \sum_{i=0}^{m+1} \binom{n}{i} \leq 2^k \cdot \sum_{i=0}^{m+1} \binom{n}{i} \leq 2^n \Rightarrow$$

$$\Rightarrow 2^{n-m \cdot s} \cdot \sum_{i=0}^{m+1} \binom{n}{i} \leq 2^n \Rightarrow \sum_{i=0}^{m+1} \binom{n}{i} \leq 2^{m \cdot s}$$

Pero esto contradice la hipótesis inicial; luego, necesariamente $d_{\mathcal{B}} = \delta$.

□

Ejemplo 3.1.2. *Probemos esta propiedad en el ejemplo 3.1.1, donde se tenía que $n = 7$, lo cual implica que $s = 3$, ya que $7 = 2^3 - 1$. Entonces, se comprueba que:*

- *Si $m = 1$, es decir, $\delta = 3$; entonces* $2^{3 \cdot 1} = 8 < \binom{7}{0} + \binom{7}{1} + \binom{7}{2} =$
  $= 1 + 7 + 21 = 29$

- *Si $m = 2$, esto es, $\delta = 5$; entonces* $2^{3\cdot 2} = 2^6 = 64 \not< \binom{7}{0} + \binom{7}{1} + \binom{7}{2} + \binom{7}{3} = 1 + 7 + 21 + 35 = 64$
- *Si $m = 3$, osea, $\delta = 7$; entonces* $2^{3\cdot 3} = 2^9 = 512 \not< \binom{7}{0} + \binom{7}{1} + \binom{7}{2} + \binom{7}{3} + \binom{7}{4} = 1 + 7 + 21 + 35 + 35 = 99$

Observando la tabla del ejemplo 3.1.1 corroboramos que para el caso $\delta = 3$ efectivamente la distancia mínima coincide con la diseñada ($d_{\mathbb{B}} = 3 = \delta$). En cambio, para los otros dos casos es evidente que no se puede aplicar el teorema 3.1.3, pues no se cumple la hipótesis; así que no se sabe si las distancias mínima y diseñada coinciden o no. De hecho, recurriendo de nuevo a la tabla, para $\delta = 5$ no se cumple que $d_{\mathcal{B}} = \delta$; mientras que para $\delta = 7$, sí. $\triangle$

3.2. Decodificación y corrección de errores en los códigos BCH

Ya hemos señalado en el apartado dedicado a los códigos cíclicos que si la distancia mínima de un código es d, entonces dicho código corrige $\left\lfloor \frac{d-1}{2} \right\rfloor$ errores.

Así, en el ejemplo 3.1.1 el código $\mathcal{B}_2(7, 5, \omega)$ corregiría $\left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{7-1}{2} \right\rfloor = \frac{6}{2} = 3$ errores.

A continuación vamos a exponer un método directo de corrección de errores para un caso algo particular. Estamos hablando de los códigos BCH de la forma $\mathcal{B}_2(2^s - 1, 5, \omega)$, que también pueden escribirse como:

$$\mathcal{B}_2(2^s - 1, 5, \omega) = \{p(x) \in A_{q,n} : p(\omega) = p(\omega^3) = 0\}$$

Tomando la distancia mínima de este código como la distancia diseñada, es decir, $d_{\mathcal{B}} = 5$ (lo cual se puede comprobar que se cumple para $s \geq 4$ [7] (pág. 360)), se tiene que el código corrige $\left\lfloor \frac{5-1}{2} \right\rfloor = \frac{4}{2} = 2$ errores. Por tanto, se puede suponer que como mucho se han producido dos errores en la transmisión del mensaje.

Sea $c(x) \in \mathcal{B}$ la palabra del código que se ha enviado y sea $u(x)$ la palabra que se ha recibido. De esta manera:

$$u(x) = c(x) + e(x), \qquad w(e(x)) \leq 2$$

Y así, hay tres posibilidades:

1. Que no se produzca ningún error, es decir, que $e(x) = 0$.
2. Que se produzca un único error en la coordenada i, esto es, $e(x) = x^i$.
3. Que se produzcan dos errores, uno en la coordenada i y otro en la coordenada j; osea que $e(x) = x^i + x^j$, para $i \neq j$.

Si $u_1 = u(\omega)$ y $u_3 = u(\omega^3)$, entonces $u_1 = e(\omega)$ y $u_3 = e(\omega^3)$ y las tres posibilidades anteriores se traducen en:

1. No se comete ningún error $\Leftrightarrow e(\omega) = e(\omega^3) = 0 \Leftrightarrow u_1 = u_3 = 0$.

2. Si se comete un único error, entonces $e(x) = x^i$ y por lo tanto $u_3 = e(\omega^3) = (\omega^3)^i = (\omega^i)^3 = (e(\omega))^3 = u_1^3 \neq 0$.

   Y recíprocamente, si $u_3 = u_1^3 \neq 0$ veamos que sólo se cometió un error. Si por el contrario, si $e(x) = x^i + x^j$, con $i \neq j$ entonces:

   $$\omega^{3i} + \omega^{3j} = e(\omega^3) = e(\omega)^3 = (\omega^i + \omega^j)^3 = \omega^{3i} + \omega^{2i}\cdot\omega^j + \omega^i\cdot\omega^{2j} + \omega^{3j} \Rightarrow$$

   $$\Rightarrow \omega^{2i}\cdot\omega^j + \omega^i\cdot\omega^{2j} = 0 \Rightarrow \omega^i\cdot\omega^j\cdot(\omega^i + \omega^j) = 0 \Rightarrow \omega^i + \omega^j = 0,\ i \neq j\sharp$$

3. Si se cometen dos errores, entonces $e(x) = x^i + x^j$, para $i \neq j$. Así que $e(\omega) = \omega^i + \omega^j = u_1$ y, además, $e(\omega^3) = \omega^{3i} + \omega^{3j} = u_3$. Poniendo $X_1 = \omega^i$ y $X_2 = \omega^j$ nos queda el siguiente sistema:

   $$\begin{cases} X_1 + X_2 = u_1 \\ \\ X_1^3 + X_2^3 = u_3 \end{cases}$$

   Se define el *Polinomio Localizador de errores* como:

   $$L(x) := (1 - X_1\,x)(1 - X_2\,x) = 1 - (X_1 + X_2)\,x + X_1\,X_2\,x^2$$

   De forma que sus raíces son X_1^{-1} y X_2^{-1}. Y si nos fijamos:

   $$u_3 = (X_1^3 + X_2^3) = (X_1 + X_2)(X_1^2 + X_1\,X_2 + X_2^2) = u_1\,(u_1^2 + X_1\,X_2) \Rightarrow$$

   $$\Rightarrow X_1\,X_2 = \frac{u_3}{u_1} - u_1^2 \Rightarrow \boxed{L(x) = 1 - u_1\,x + (\tfrac{u_3}{u_1} - u_1^2)\,x^2}$$

   Así que, finalmente, basta calcular las raíces de $L(x)$ evaluándolo en los $2^s - 1$ elementos no nulos de $\mathbb{F}_{2^s}$ para obtener las coordenadas en las que se han producido errores.

En conclusión, tenemos el Algoritmo recogido en el siguiente teorema:

Teorema 3.2.1. *Sea $\mathcal{B} = \mathcal{B}_2(2^s - 1, 5, \omega)$, con $s \geq 4$ y supongamos que al recibir la palabra $u(x)$ se cometieron a lo sumo dos errores. Sea $u_1 = u(\omega)$ y $u_3 = u(\omega^3)$. Entonces:*

1. *Si $u_1 = u_3 = 0$, no hay error.*

2. *Si $u_3 = u_1^3 \neq 0$, hay un error; y si $u_1 = \omega^i$, entonces el error está en la coordenada i.*

3. Si $u_1 \neq 0$ y $u_3 \neq u_1^3$, hay dos errores. El polinomio localizador de errores *sería $L(x) = 1 - u_1 \cdot x + (\frac{u_3}{u_1} - u_1^2) \cdot x^2$, que tiene dos raíces distintas ω^{n-i} y ω^{n-j} y los errores se encuentran en las coordenadas i y j.*

Ejemplo 3.2.1. *Sea $n = 15 = 2^4 - 1$. De forma que, $SF(x^{15}-1)_{\mathbb{F}_2} = \mathbb{F}_{2^{o_n(2)}} = \mathbb{F}_{2^4} = \mathbb{F}_{16}$. Podemos tomar $\mathbb{F}_{16} = \frac{\mathbb{F}_2[x]}{<x^4+x+1>}$, donde $x^4 + x + 1$ es un polinomio primitivo.*

Sea ω una raíz primitiva; de forma que, $\omega^4 + \omega + 1 = 0$ y $\mathbb{F}_{16}^* =< \omega >$.

Consideremos el código $\mathcal{B} = \mathcal{B}_2(15, 5, \omega)$. Los ceros de $\mathcal{B}$ son $\omega, \omega^2, \omega^3, \omega^4$. Así, los conjuntos ciclotómicos son:

$$C_0 = \{0\}$$

$$C_1 = \{1, 2, 4, 8\}$$

$$C_3 = \{3, 6, 9, 12\}$$

$$C_5 = \{5, 10\}$$

$$C_7 = \{7, 14, 13, 11\}$$

Y así, con algunos cálculos los polinomios que necesitaremos son:

$$m_1(x) = \prod_{i \in C_1} (x - \omega^i) = x^4 + x + 1$$

$$m_3(x) = \prod_{j \in C_3} (x - \omega^j) = x^4 + x^3 + x^2 + x + 1$$

Y esto nos lleva a que $\mathcal{B} =\ll m_1(x) \cdot m_3(x) \gg =\ll x^8 + x^7 + x^6 + x^4 + 1 \gg$

Pues bien, supongamos que queremos enviar la palabra $c(x) = m_1(x) = x^4 + x + 1 \in \mathcal{B}$. Si suponemos además que en la transmisión del mensaje se cometen dos errores en las coordenadas 4 y 12, tendríamos que $e(x) = x^4 + x^{12}$ y que la palabra recibida sería $u(x) = x^{12} + x^8 + x^7 + x^6 + 1$. Con el fin de aplicar el teorema 3.2.1, hallamos:

$$u_1 = u(\omega) = \omega^{12} + \omega^8 + \omega^7 + \omega^6 + 1 = \omega^6$$

$$u_3 = u(\omega^3) = \omega^{36} + \omega^{24} + \omega^{21} + \omega^{18} + 1 = \omega^4$$

Como $u_1 \neq 0$ y $u_1^3 = (\omega^6)^3 = \omega^{18} = \omega^3 \neq u_3$, esto nos lleva a confirmar que hay dos errores. Por ello, armando el polinomio localizador de errores recuperaríamos la palabra del código:

$$L(x) = 1 + \omega^6 \cdot x + \left(\frac{\omega^4}{\omega^6} + \omega^{12}\right) \cdot x^2 = 1 + \omega^6 \cdot x + (\omega^{-2} + \omega^{12}) \cdot x^2$$

Ahora, evaluando $L(x)$ en $1, \omega, ..., \omega^{14}$, hallamos sus raíces, que son ω^3 y ω^{11}. Y como $\omega^3 = \omega^{15-12}$ y $\omega^{11} = \omega^{15-4}$, los errores están en las coordenadas 4 y 12, como ya sabíamos desde el principio.$\triangle$

El Caso General

El algoritmo anteriormente expuesto se puede generalizar a todos los códigos BCH binarios. Veámoslo brevemente, si $u(x) = c(x) + e(x)$ es la palabra recibida entonces podemos determinar fácilmente las cantidades:

$$u_1 = e(\omega); u_3 = e(\omega^3) \ ... \ u_{\delta-2} = e(\omega^{\delta-2})$$

Si suponemos que los errores han ocurrido en las localizaciones $i_1, ..., i_l$ en $e(x)$, entonces:

$$e(x) = \sum_{j=1}^{l} x^{i_j}$$

Así que, sustituyendo:

$$\sum_{j=1}^{l} \omega^{i_j} = u_1; \qquad \sum_{j=1}^{l} \omega^{3i_j} = u_3 \qquad ... \qquad \sum_{j=1}^{l} \omega^{(\delta-2)\, i_j} = u_{\delta-2}$$

A continuación, definimos los localizadores del error como $X_j = \omega^{i_j}$, los cuales nos indican las posiciones de los errores. Así, el sistema al que habíamos llegado antes, en esta ocasión quedaría:

$$\sum_{j=1}^{l} X_j = u_1; \qquad \sum_{j=1}^{l} X_j^3 = u_3 \qquad ... \qquad \sum_{j=1}^{l} X_j^{(\delta-2)} = u_{\delta-2}$$

Y el polinomio localizador de errores es:

$$L(x) = \prod_{j=1}^{l} (1 - X_j \, x)$$

Cuyas raíces son las inversas de los localizadores X_j.

En el caso más particular que vimos previamente, teníamos $l = 2$, por lo que según lo que acabamos de ver tendríamos el sistema:

$$L(x) = \prod_{j=1}^{2} (1 - X_j \, x) = (1 - X_1 \, x)(1 - X_2 \, x) = 1 - (X_1 + X_2) \, x + X_1 \, X_2 \, x^2$$

que concide justamente con el polinomio localizador que ya habíamos definido.

Una vez hallados los coeficientes del polinomio localizador, las soluciones que buscamos se pueden determinar por intento y error, siempre y cuando el cuerpo en que nos movamos no sea demasiado grande.

3.3. Códigos Reed-Solomon

Como cabía esperar, los Códigos Reed-Solomon también deben su nombre a sus inventores I.Reed y G.Solomon (1960). Además, como ya adelantamos en la sección anterior,

este tipo de códigos conforman un caso particular de los Códigos BCH. Su utilidad vio su esplendor en 1968-1969, cuando fueron usado como algoritmo de decodificación bajo el nombre Berlekamp-Massey. Incluso actualmente, los códigos Reed-Solomon forman parte de numerosas aplicaciones tecnológicas. Por ejemplo, son empleados en el almacenamiento de datos (casete, DVD, códigos de barra...), en las comunicaciones inalámbricas (móviles...), en las comunicaciones satélites, en la televisión digital, etc. Su característica más importante es que permiten fijar a priori la distancia.

En primer lugar, recordemos la definición:

Definición 3.3.1. *Sea $q \geq 3$, un **Código Reed-Solomon** q-ario es un Código BCH de longitud $n = q - 1$. Esto significa que:*

$$\mathcal{RS}_q(\delta, \omega, b) = \mathcal{B}_q(q-1, \delta, \omega, b)$$

Pues bien, como $o_{q-1}(q) = 1$, entonces SF $(x^{q-1} - 1) = \mathbb{F}_q$ y:

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

Y de esta forma apreciamos que si ω es un elemento primitivo de $\mathbb{F}_q$, entonces el polinomio mínimo de ω^i es $m_{\omega^i} = x - \omega^i \in \mathbb{F}_q[x]$. Es decir, $C_i = \{i\}$ para $i = 1, 2, ..., n-1$ y:

$$\mathcal{RS}_q(\delta, \omega, b) = \ll (x - \omega^b) \cdot (x - \omega^{b+1}) ... (x - \omega^{b+\delta-2}) \gg$$

Por otra parte, la dimensión de un código Reed-Solomon es:

$$k = dim(\mathcal{RS}_q(\delta, \omega, b)) = n - deg(g(x)) = n - (\delta - 1)$$

Mientras que la distancia es:

$$d = \delta = n - k + 1$$

Lo cual es fácil de ver, pues por la cota BCH ocurre que $d \geq \delta = n - k + 1$, pero por la cota de Singleton, se cumple que $d \leq n - k + 1$; luego necesariamente se da la igualdad.

Resumiendo, los códigos Reed-Solomon tienen distancia mínima δ y son códigos MDS (*Maximum Distance Separable*), dado que $k = n - d + 1$.

Ejemplo 3.3.1. *Sea $q = 4$ y $b = 1$, de manera que $n = 4 - 1 = 3$ y SF $(x^{q-1} - 1)_{\mathbb{F}_q} =$ SF $(x^3 - 1)_{\mathbb{F}_q} = \mathbb{F}_4 = \{0, 1, \omega, \omega^2\} = \{0, 1, \omega, \omega + 1\}$. Por consiguiente:*

$$\mathcal{RS}_4(3, \omega) = \mathcal{RS}_4(3, \omega) = \ll (x - \omega)(x - \omega^2) \gg = \ll x^2 + x + 1 \gg$$

Mientras que:

$$\mathcal{RS}_4(3, \omega^2) = \ll (x - \omega^2)(x - \omega^3) \gg = \ll (x - \omega^2) \cdot (x - 1) \gg = \ll x^2 + \omega \cdot x + \omega^2 \gg$$

Asimismo, se ve claramente que los dos códigos Reed Solomon expuestos tienen parámetros $[n, k, d] = [3, 1, 3]$.

Finalizamos con un teorema que recoge una propiedad interesante de los códigos Reed-Solomon:

Teorema 3.3.1. *El dual de un código Reed-Solomon es también un código Reed-Solomon.*

Demostración. Sea $\mathcal{C}$ un código cíclico con polinomio generador $g(x)$ y polinomio de control $h(x)$; y $T = \cup_s C_s$ la unión de los conjuntos ciclotómicos. Notemos que, usando esta notación:

$$\mathcal{Z} = \{\alpha^i \ / \ i \in T\}$$

Este conjunto T se conoce como *conjunto definidor de* $\mathcal{C}$.

Si ahora $N = \{0, 1, ..., n-1\}$, se tiene que $N \setminus (-1)T \ (mod\ n)$ es el conjunto definidor de $\mathcal{C}^\perp = \ll g^\perp(x) \gg$. Para ver este resultado, recordemos que $x^n - 1 = g(x) \cdot h(x)$ entonces las raíces de $x^n - 1$ que no sean raíces de $g(x)$ lo serán de $h(x)$, o sea que $h(\alpha^j) = 0$ con $j \in N \setminus T \ (mod\ n)$. Asimismo, por el teorema 2.2.1 sabemos que $g^\perp(x) = h_0^{-1}\, x^{n-r}\, h(x^{-1})$ por lo que los ceros del código dual son:

$$\mathcal{Z}^\perp = \{\alpha^{-j} \ / \ j \notin T\} = \{\alpha^j \ / \ -j \notin T\} = \{\alpha^j \ / \ j \notin (-1)T\} = \{\alpha^j \ / \ j \in N \setminus (-1)T\}$$

Concluyendo que el cojunto definidor de $\mathcal{C}^\perp$ es $N \setminus (-1)T \ (mod\ n)$, como ya decíamos.

En el caso concreto de los códigos Reed-Solomon de longitud n y distancia diseñada δ se tiene que $T = \{b, b+1, ..., b+\delta-2\}$, es decir que T es un conjunto de $\delta - 1$ elementos consecutivos de N. Se observa que entonces $(-1)T \ (mod\ n)$ es también un conjunto de $\delta - 1$ elementos consecutivos de N. De esta manera, se tiene que $N \setminus (-1)T \ (mod\ n)$ es un conjunto de $n - \delta + 1$ elementos consecutivos; lo cual demuestra que el dual $\mathcal{C}^\perp$ de un código Reed-Solomon $\mathcal{C}$ es un código Reed-Solomon.

□

Ejemplo 3.3.1. *Tomemos* $\alpha = 2$ *que sabemos que es un elemento primitivo en* $\mathbb{F}_{13}$*. Sea* $\mathcal{C}$ *un código Reed-Solomon en el sentido estricto (*$b = 1$*), con distancia diseñada* $\delta = 5$ *en* $\mathbb{F}_{13}$ *y longitud* $n = 12$*. Es decir, un* $[12, 8, 5]$*-código Reed-Solomon. Hallemos su código dual:*

El conjunto definidor de $\mathcal{C}$ *sería* $T = \{1, 2, 3, 4\}$ *y el polinomio generador* $(x-2)(x-2^2)(x-2^3)(x-2^4) = x^4 + 9x^3 + 7x^2 + 2x + 10$*. Por consiguiente,* $(-1)T \ (mod\ 12) = \{-1, -2, -3, -4\} \ (mod\ 12) = \{11, 10, 9, 8\} \ (mod\ 12)$ *lo cual implica que el conjunto definidor de* $\mathcal{C}^\perp$ *es* $N \setminus (-1)T \ (mod\ 12) = \{0, 1, 2, ..., 12\} \setminus \{11, 10, 9, 8\} = \{0, 1, 2, 3, 4, 5, 6, 7\}$*. Esto nos lleva a que el polinomio generador del código dual es* $g^\perp(x) = x^8 + 5x^7 + 10x^6 + 4x^5 + 11x^4 + 5x^3 + x^2 + 12x + 3$*. Concluyendo que* $\mathcal{C}^\perp = \ll x^8 + 5x^7 + 10x^6 + 4x^5 + 11x^4 + 5x^3 + x^2 + 12x + 3 \gg$ *es un* $[12, 4, 9]$*-código Reed-Solomon.*

Por último, podemos observar que un código Reed-Solomon tiene capacidad correctora $t = \left\lfloor \frac{d-1}{2} \right\rfloor = \left\lfloor \frac{\delta-1}{2} \right\rfloor$, pues ya demostramos que $d = \delta$.

Capítulo 4

Aplicación: Códigos Reed-Solomon y CD's

Los códigos Reed-Solomon tienen sus principales aplicaciones en áreas relacionadas con el almacenamiento de información y su posterior reproducción (CD's, DVD's, videojuegos, ...), con la telefonía móvil o con las comunicaciones en general.

Seguro que muchos se han preguntado alguna vez cómo es posible que, a pesar de que la superficie de un CD esté seriamente dañada (manchada, o con huellas dactilares, por ejemplo), la calidad del sonido siga siendo alta o, dicho de otra forma, que podamos recuperar la información que tiene almacenada.

La clave está en que gran parte de esta información es redundante y esto se consigue usando códigos potentes que añaden datos para que, si se producen errores en el almacenamiento o en la transmisión, el decodificador los pueda detectar y corregir.

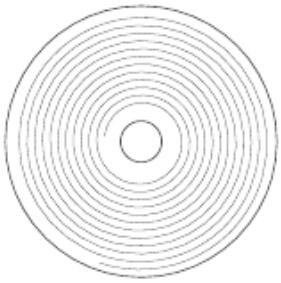
En este capítulo, vamos a dar una idea aproximada y tal vez un tanto simplificada de lo que se esconde tras un CD.

El sistema de audio digital que conocemos como *Compact Disc (CD)* fue desarrollado conjuntamente por las compañías N.V. Philips y Sony Corporation en 1979.

Un CD es un disco de aluminio de 120 milímetros de diámetro y cubierto con una capa de plástico. En realidad, se trata de una larguísima espiral de 5 kilómetros de longitud sobre la que se graba una sucesión de depresiones llamadas "valles" o "*pits*" (de 0,12μm de profundidad) y "llanuras" o "*lands*"; ambas grabaciones representarán ceros y unos para el lector láser, que tiene una longitud de onda aproximada de 0,8μm y una velocidad constante de 1,25m/s. El proceso de lectura se basa en que la luz del láser se refleja con distintas intensidades en los valles y las llanuras.

Como hemos comentado, los datos contenidos en estos valles y llanuras están sujetos a errores debidos a partículas en el disco, manchas, huellas dactilares, o burbujas de aire que en la cubierta plástica. Por lo general, se trata de errores en ráfaga, es decir, errores consecutivos en varias componentes de las palabras código, que a menudo se repiten de forma continuada.

Se muestra a continuación una imagen esquemática de lo que sería el CD:



4.1. Codificación y grabado

En este apartado, vamos a describir cómo se codifican los datos de audio para grabarlos en el CD.

El primer paso es convertir las ondas sonoras analógicas en digitales usando un proceso de muestreo que asigna una cadena binaria de longitud 16 a cada intervalo de tiempo.

Por otra parte, ya que el sonido se va a reproducir en estéreo, en realidad se toman dos muestras a la vez, una para el canal izquierdo y otra para el derecho. De esta forma, cada muestra produce dos vectores de $\mathbb{F}_2^{16}$, uno para cada canal. Si cada uno de estos vectores se corta a la mitad, se obtiene un elemento del cuerpo $\mathbb{F}_{2^8}$, esto es un *byte* y así, cada muestra produce cuatro *bytes* de datos.

Para la grabación de CD las ondas de sonido se convierten normalmente con un ratio de 44,1 kHz , esto es a razón de 44100 pares de muestras por segundo (para audio profesional puede ser superior, pero un ratio superior a 50, 60 kHz no añade información suplementaria que sea perceptible por el oido humano). De esta forma, por cada segundo de sonido grabado, se generan $44100 \cdot 32 = 1411200$ *bits*, que equivalen a 176400 *bytes*.

El siguiente paso es codificar esos *bytes*, y para ello se requiere el uso de dos códigos Reed-Solomon, $\mathcal{C}_1$ y $\mathcal{C}_2$, y dos formas de intercalado cruzado (*Cross-Interleaved*), cuyo propósito es romper los errores en ráfaga. Esta combinación se conoce como *CIRC* (Cross-Interleaved Reed-Solomon Code).

A grandes rasgos, el proceso que se sigue consiste en tomar datos, entrelazarlos, codificar los datos con $\mathcal{C}_1$, entrelazar de nuevo la salida y codificar ahora con $\mathcal{C}_2$. Los parámetros de $\mathcal{C}_1$ son $[28, 24, 5]$; mientras que los de $\mathcal{C}_2$ son $[28, 32, 5]$, esto es, ambos códigos añaden 4 símbolos de paridad.

Para el primer entrelazado, se agrupan los datos en bloques de 24 *bytes*, formados por seis muestras de cuatro *bytes* cada una.

$$\begin{array}{c} L_1R_1L_2R_2L_3R_3L_4R_4L_5R_5L_6R_6 \\ \widehat{L_1}\widehat{R_1}\widehat{L_2}\widehat{R_2}\widehat{L_3}\widehat{R_3}\widehat{L_4}\widehat{R_4}\widehat{L_5}\widehat{R_5}\widehat{L_6}\widehat{R_6} \\ \widetilde{L_1}\widetilde{R_1}\widetilde{L_2}\widetilde{R_2}\widetilde{L_3}\widetilde{R_3}\widetilde{L_4}\widetilde{R_4}\widetilde{L_5}\widetilde{R_5}\widetilde{L_6}\widetilde{R_6} \\ \vdots \end{array}$$

Donde los L_i están compuestos por dos *bytes* para el canal izquierdo correspondiente

a la muestra i-ésima, y los R_i constan de dos *bytes* para el canal derecho de la muestra i-ésima. Al aplicar el codificador se produce un retardo de dos símbolos entre las muestras pares e impares, es decir, se agrupan las muestras impares, L_1R_1, L_3R_3 y L_5R_5, con los símbolos pares que están en dos bloques posteriores.

$$L_1R_1\widetilde{L}_2\widetilde{R}_2L_3R_3\widetilde{L}_4\widetilde{R}_4L_5R_5\widetilde{L}_6\widetilde{R}_6$$

A continuación, reordenamos las muestras separando las muestras pares y las impares y a su vez los canales derecho e izquierdo y queda una nueva estructura:

$$L_1L_3L_5R_1R_3R_5\widetilde{L}_2\widetilde{L}_4\widetilde{L}_6\widetilde{R}_2\widetilde{R}_4\widetilde{R}_6$$

La combinación de entrelazado y retardo busca separar los datos de forma que se pueda recuperar la información, incluso con errores en ráfaga que produzcan la pérdida de dos bloques consecutivos. Además, la permutación de pares e impares ayuda a ocultar errores.

El bloque de 24 *bytes* es un elemento de $\mathbb{F}_{256}^{24}$ y se codifica usando $\mathcal{C}_1$, que produce cuatro *bytes* de redundancia, esto es, dos pares P_1 y P_2 cada uno con dos *bytes* de paridad que se colocan en el centro del bloque que ya teníamos, para separar aún más las muestras pares y las impares:

$$L_1L_3L_5R_1R_3R_5P_1P_2\widetilde{L}_2\widetilde{L}_4\widetilde{L}_6\widetilde{R}_2\widetilde{R}_4\widetilde{R}_6$$

El siguiente paso es la intercalación cruzada de las palabras del código. En este caso, cada *byte* se retrasa en periodos diferentes siguiendo el esquema representado por la matriz:

$$\begin{pmatrix} c_{1,1} & c_{2,1} & c_{3,1} & c_{4,1} & c_{5,1} & c_{6,1} & c_{7,1} & c_{8,1} & c_{9,1} & c_{10,1} & \dots \\ 0 & 0 & 0 & 0 & c_{1,2} & c_{2,2} & c_{3,2} & c_{4,2} & c_{5,2} & c_{6,2} & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1,3} & c_{2,3} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

donde la palabra i-ésima del código es $c_i = c_{i,1}c_{i,2}...c_{i,28}$

Los 28 *bytes* de cada palabra del código quedan ahora distribuidos en bloques diferentes y al codificar con $\mathcal{C}_2$, se obtienen palabras de 32 *bytes*. Por último, se introduce un nuevo entrelazando reagrupando los términos pares de una palabra del código con los impares de la siguiente y la cadena que resulta se divide en segmentos de 32 *bytes* (donde 16 *bytes* son de una palabra de $\mathcal{C}_2$ y los 16 restantes de otra, debido al reagrupamiento realizado). Además, se añade un último *byte* que contiene control e información del CD (tiempo de duración, compositor, etc.).

El último paso en esta fase es la impresión o grabado de los datos en el CD, y para ello se deben convertir los *bits* de datos en *bits* de canal, que son los realmente están impresos en el disco y se decodifican cuando se reproduce.

Los valles y llanuras se representan por ceros, mientras que cada paso de entre éstos se representa con un 1. La grabación de cada *bit* en el CD ocupa 0,3μm de longitud en cada pista, así que la cadena 10000000100 representa un valle de longitud 2,4μm , seguido de una llanura de longitud 0,9μm.

La conversión natural de *bytes* a *bits*, es decir, la conversión de cada *byte* en cadenas binarias de 8 *bits*, o elemntos de $\mathbb{F}_2^8$, no se puede hacer por distintas razones técnicas (como la separación máxima y mínima que debe haber entre los 0s y los 1s de cada cadena) y la longitud mínima para la que podemos encontrar 256 cadenas diferentes que cumplan estas condiciones es 14. Esta conversión se hará entonces por un método que se conoce como *EFM*. Además, hay que añadir 3 *bits* adicionales de fusión (para que al unirlas sigan cumpliendo las condiciones de separación). De esta forma, cada bloque de 24 *bytes* se transforma en 33 después del proceso de *CIRC*, y cada uno de éstos produce 17 *bits*. En definitiva, cada bloque seis muestras (24 *bytes*) produce $33 \cdot 17$ *bits*, a los que se añaden 24 de sincronización y 3 de fusión; luego, cada estructura de seis muestras conduce a 588 *bits* (si 1 seg suponía 176,400 *bytes*, entonces cada segundo es 176,400 *bytes*, osea, $176{,}400/24 = 7350$ bloques de 24 *bytes* que producen $7350 \cdot 588 = 4{,}321{,}800$ *bits* $= 4,321$ Mbps).

4.2. Decodificación y corrección de errores

Cuando se reproduce el disco , el sistema de corrección de errores elimina los *bits* de sincronización, los de paridad y de fusión y usa un proceso llamado *demodulación* para recuperar cadenas de 32 *bytes* a partir de los datos proporcionados por el *EFM (Eight-to-Fourteen Modulation)*. Una vez se ha realizado, se deben desahacer los intercalados y usar un decoficador Reed-Solomon $\mathcal{C}_2$. El decodificador analiza los datos y comprueba si los símbolos de paridad son correctos o si se ha producido algún error y, en este caso, usa el síndrome para localizar el error. Las palabras incorrectas se marcan como recuperables (si se pueden corregir), irrecuprable o posiblenmente corregibles. Ahora bien, como $\mathcal{C}_2$ es un $[32, 28, 5]$-código Reed-Solomon en $\mathbb{F}_{256}$, entonces sabemos que puede corregir dos errores, y detectar 4 errores.

Sin embargo, sólo se utiliza para corregir un único error debido a que, en esta situación, la probabilidad de que detecte cuatro o más errores es mucho mayor que cuando se usa con toda su capacidad correctora.

Para ver esto, si asumimos que todas las palabras del código son igualmente parecidas, observamos que la esfera de radio 1 centrada en alguna palabra del código c_1 no contiene a ningún vector que difiera de otra palabra del código c_2 en como mucho tres posiciones, ya que c_1 y c_2 tienen como mínimo distancia 5 uno del otro, y la probabilidad de que $\mathcal{C}_2$ se equivoque al detectar cuatro o más errores es igual a la relación entre el número total de vectores dentro de la esfera de radio 1 centrada en las palabras del código y el número total de vectores en $\mathbb{F}_{256}^{32}$:

$$\frac{256^{28}[1 + 32(256 - 1)]}{256^{32}} \approx 1{,}9 \cdot 10^{-6}$$

En cambio, si $\mathcal{C}_2$ fuera utilizado con toda su capacidad correctora, esta realción es :

$$\frac{256^{28}[1 + 32(256 - 1) + \binom{32}{2}(256 - 1)^2]}{256^{32}} \approx 7{,}5 \cdot 10^{-3}$$

Si el decodificador para $\mathcal{C}_2$ determina que en una cadena de 32 *bytes* no hay errores, extrae los mensajes de 28 *bytes* de datos y pasa a la siguiente fase. Si $\mathcal{C}_2$ detecta un único error, lo corrige y de nuevo transmite el mensaje de 28 *bytes*. Por último, si detecta más de dos errores, entonces transmite una cadena de 28 *bytes* con todas las componentes marcadas. Estas cadenas de 28 *bytes* son las columnas de la matriz expuesta anteriormente y sus diagonales con pendiente $-1/4$ se pasan como vectores de 28 *bytes* por el decodificador para $\mathcal{C}_1$.

$\mathcal{C}_1$ puede corregir tanto errores en ráfaga como errores de *bits* que $\mathcal{C}_2$ no pudo corregir y cuando no puede completar la corrección (por ejemplo, si hay más de cuatro *bytes* marcados como erróneos por estructura) se marcan los 24 bytes como incorrectos y pasan al proceso de interpolación y/o sileciamiento. La etapa final exige un nuevo desentrelazado y retardo de los símbolos de las palabras pares para compensar los que se hicieron en el proceso de codificación.

En la detección y corrección de errores hay que guardar equilibrio entre el exceso de redundancia, la complejidad y coste de los codificadores y decodificadores, y su capacidad correctora. Para compensar esto, se someten los datos a distintos tratamientos que intenten ocultar los errores detectados que no se han podido no corregir y disminuir el núemro de errores que no se detectan (y que, por lo tanto, no podríamos ocultar).

Para la ocultación de errores se usa la *interpolación* que consiste en usar datos correctos para reemplazar los que contienen error. Por ejemplo, se puede repetir el valor de la última muestra correcta antes de producirse el error, o asociar al valor perdido la media aritmética realizada sobre las muestras anterior y posterior a la errónea, o usar una combinación de ambas interpolaciones.

El *silenciamiento (muting)* es el proceso reemplazar por un dato de valor cero las muestras irrecuperables o que no se pudieron corregir y trata de evitar que se produzcan chasquidos, que son mucho más perceptibles que los silencios (no se perciben si no superan el margen de 1 a 4 ms, y los algoritmos que los producen pueden graduar el silenciamiento y la recuperación de la señal de salida de audio).

Para finalizar, hay que señalar que generlamente el *CIRC* permite la corrección de hasta 3874 *bits* consecutivos, que corresponde a un defecto en la pista de $2,5$ mm. de longitud. Asimismo, la integridad de los datos contenidos en un CD se evalúa usando varios indicadores que miden el número de errores (tales como el *BLER* (BLock Error Rate) = numéro de bloques por segundo en los que hay al menos un erróneo, o el *CER* = número de errores en ráfaga) y que se tienen en cuenta en los distintos controles de calidad.

Capítulo 5

Conclusiones

Tras la lectura y análisis de varios libros, artículos y notas sobre la Teoría de Códigos, no cabe duda de su inmensidad y de la presencia en este campo de las matemáticas, en especial del Álgebra. Asimismo, y aunque no siempre nos percatemos de ello, los códigos correctores forman parte de la vida cotidiana. Por ejemplo, cuando vamos al supermercado y la dependienta pasa el artículo por el lector, o cuando trabajamos con el ordenador, o escuchamos música de un CD o vemos una película grabada en un DVD. Incluso, tras la magia podemos encontrar códigos correctores de errores ([1]).

Cualquier dispositivo de almacenamiento o canal de transmisión introduce errores en los datos. Debido a la naturaleza variada de los errores, se deben contrastar varias técnicas para proteger el mensaje (como son la adición de paridad, la redundancia, la intercalación o el entrelazado).

Por otra parte, existen muchos tipos de sistemas de codificación y decodificación de errores. Por ello, para la elección de uno de ellos conviene tener en cuenta aspectos como las posibilidades de corrección que ofrecen, el exceso de *bits* redundantes, los fallos en las detecciones, la longitud del código o el coste. Aunque un sistema ideal de detección y corrección de errores es teóricamente posible, es impracticable debido a la necesidad de transmitir gran cantidad de redundancia.

Según la bibliografía consultada, la variada naturaleza de los errores que se producen al transmitir la información hace necesario utlizar a menudo la combinación de varios códigos, así como técnicas que faciliten la corrección de errores.

La característica más importante y destacable de los códigos correctores de errores es la robustez que aportan a la transmisión y al almacenamiento de información.

Apéndice A

Factorización de $x^n - 1$ sobre un cuerpo finito $\mathbb{F}_q$

Veamos aquí de forma resumida y no muy detallada cómo factorizar $x^n - 1$ sobre cuerpos finitos. Para más información se puede consultar [3] (pág. 112). Comencemos para ello con la siguiente definición:

Definición A.1. *Sea n y q enteros positivos relativamente primos, y sea s un entero con $0 \leq s < n$. El **conjunto q-ciclotómico de s módulo** n viene dado por:*

$$C_s = \{s, s\,q, ..., s\,q^{k-1}\}\ (mod\, n)$$

donde k es el menor entero positivo tal que $s\,q^{k-1} \equiv s\ (mod\, n)$.

De esta descripción, se sigue que los conjuntos ciclotómicos módulo n particionan el conjunto de los enteros $\{0, 1, ..., n-1\}$. Normalmente denotamos un conjunto ciclotómico en esta partición eligiendo s de forma que sea el menor entero incluido en él.

Ahora, a través del siguiente teorema, vamos a ver que existe una relación entre conjuntos ciclotómicos módulo n y el conjunto polinomios mínimos en $\mathbb{F}_q$ de las raíces n-ésimas de la unidad.

Teorema A.1. *Sea $\mathbb{F}_{q^t}$ una extensión del cuerpo $\mathbb{F}_q$, y sea α un elemento de $\mathbb{F}_{q^t}$. Sea además n el menor entero tal que $\alpha^n = 1$ (es decir, α es una raíz n-ésima primitiva de la unidad en $\mathbb{F}_{q^t}$). entonces:*

1. *Para cada entero s, con $0 \leq s < n$, el polinomio mínimo de α^s en $\mathbb{F}_q$ es:*

$$M_{\alpha^s}(x) = \prod_{i \in C_s} (x - \alpha^i)$$

   donde C_s es el conjunto q-ciclotómico de s módulo n.

2. *En particular, los conjugados de α son los elementos α^i, con $i \in C_1 = \{1, q, ..., q^{k-1}\}$*

3.

$$x^n - 1 = \prod_s M_{\alpha^s}(x)$$

donde s recorre un conjunto de representates de los conjuntos q-ciclotómicos de s módulo n.

Veamos un ejemplo sencillo:

Ejemplo A.1. *Supongamos que queremos factorizar en irreducibles el polinomio $x^7 - 1$ en $\mathbb{F}_2[x]$. Vamos a considerar la extensión $\mathbb{F}_{2^3} = \mathbb{F}_8$ (es decir $n = 7$, $q = 2$ y $t = 3$), siendo entonces α una raíz primitiva séptima de la unidad en dicha extensión de cuerpo.*

Pues bien, como $n = 7$, entonces $s \in [0, 7)$ y, por lo tanto, los conjuntos ciclotómicos serán:

- $C_0 = \{0\}$
- $C_1 = \{1, 1\cdot 2, 1\cdot 2^2\} = \{1, 2, 4\} (mod\, 7)$
- $C_2 = \{2, 2\cdot 2, 2\cdot 2^2\} = \{2, 4, 1\} (mod\, 7)$
- $C_3 = \{3, 3\cdot 2, 3\cdot 2^2\} = \{3, 6, 5\} (mod\, 7)$
- $C_4 = \{4, 4\cdot 2, 4\cdot 2^2\} = \{4, 1, 2\} (mod\, 7)$
- $C_5 = \{5, 5\cdot 2, 5\cdot 2^2\} = \{5, 3, 6\} (mod\, 7)$
- $C_6 = \{6, 6\cdot 2, 6\cdot 2^2\} = \{6, 5, 3\} (mod\, 7)$

De modo que, en definitiva:

- $C_0 = \{0\}$
- $C_1 = C_2 = C_4 = \{1, 2, 4\}$
- $C_3 = C_5 = C_6 = \{3, 6, 5\}$

Y esto nos lleva a los polinomios mínimos:

- $M_1 = \prod_{i \in C_0}(x - \alpha^i) = x + 1$
- $M_\alpha(x) = M_{\alpha^2}(x) = M_{\alpha^4}(x) = \prod_{i \in C_1}(x - \alpha^i) = (x - \alpha)(x - \alpha^2)(x - \alpha^4) =$
  $= x^3 + (\alpha^2 + \alpha + \alpha^4)\cdot x^2 + (\alpha^3 + \alpha^6 + \alpha^5)\cdot x + 1$
- $M_{\alpha^3}(x) = M_{\alpha^5}(x) = M_{\alpha^6}(x) = \prod_{i \in C_3}(x - \alpha^i) = (x - \alpha^3)(x - \alpha^5)(x - \alpha^6) =$
  $= x^3 + (\alpha^6 + \alpha^3 + \alpha^5)\cdot x^2 + (\alpha + \alpha^2 + \alpha^4)\cdot x + 1$

Ahora bien, como no tenemos el valor de α, para conocer quiénes son exactamente los polinomios anteriores tenemos en cuenta que en $\mathbb{F}_2[x]$:

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

Por lo que nuestro α debe cumplir que:

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

Lo cual nos abre camino a dos posibilidades:

1. $\alpha + \alpha^2 + \alpha^4 = 0$ *(es decir,* $\alpha^3 + \alpha + 1 = 0$ *y* $\mathbb{F}_{2^3} \cong \frac{F_2[x]}{x^3+x+1}$*), en cuyo caso:*
   - $M_\alpha(x) = x^3 + x + 1$
   - $M_{\alpha^3}(x) = x^3 + x^2 + 1$
2. $\alpha^3 + \alpha^5 + \alpha^6 = 0$ *(esto es,* $\alpha^3 + \alpha^2 + 1 = 0$ *y* $\mathbb{F}_{2^3} \cong \frac{F_2[x]}{x^3+x^2+1}$*). Si ésta fuera la situación, ocurriría que:*
   - $M_\alpha(x) = x^3 + x^2 + 1$
   - $M_{\alpha^3}(x) = x^3 + x + 1$

En cualquier caso la factorización en irreducibles de $x^7 - 1$ en $\mathbb{F}_2[x]$:

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

Bibliografía

[1] Pedro Alegría. Códigos secretos y teoría de la información en la magia. *Depertamento de Matemáticas Universidad del País Vasco*, Noviembre 2004. www.ehu.es/ mtpalezp/mates/codigos.pdf.

[2] Irene Márquez Corbella. Introducción a los aspectos geométricos-algebraicos de la teoría de códigos. *Trabajo académicamente dirigido (Universidad de La Laguna)*, Junio 2008.

[3] W. Cary Huffman and Vera S.Pless. *Fundamentals of Error-correcting Codes.* Cambridge University Press, 2003.

[4] Ricardo A. Podestá. Códigos cíclicos. *Jornadas de Criptografía y Códigos Autocorrectores (JCCA)*, Noviembre 2006. http://www.famaf.unc.edu.ar/ cripto06.

[5] Ken C. Pohlmann. *Principios de audio digital.* McGraw-Hill, 2002.

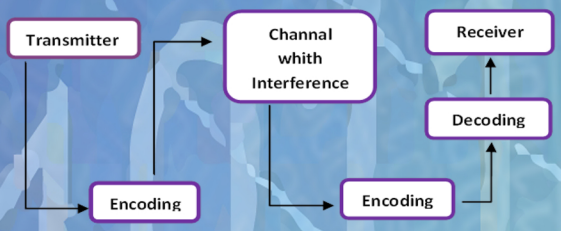
[6] Mª Victoria Sánchez Reyes. Notas sobre códigos correctores. *Mátematicas de las Comunicaciones (Máster en Matemática Avanzadas y Aplicaciones)(ULL)*, Febrero 2013.

[7] Steven Roman. *Coding and Information Theory.* Graduate Texts in Mathematics 134, Springer, 1992.

[8] W. Cary Huffman ; Vera S.Pless and Richard A. Brualdi. *An Introduction to Algebraic Codes.* Handbook of Coding Theory, 1998.

Coding theory try to find efficient codes to detect and correct errors that could have occurred during the transmission of a message. The aim of this presentation is being an introduction to Error – Correcting codes and show how Mathematics is behind of this theory. For some specific case we will be able to answer the question: How could we know is some information received is corrected? Or how could we recover original data?

MESSAGE TRANSMISSION PROGRESS



Let C be a Code
Between properties of codes we can find:

- Hamming distance:
$$d(C) = d_H(C) = \min\{d_H(x,z): x,z \in C, x \neq z\}$$
- Weight:
$$w(C) = \min\{d_H(x,0): x \in C\}$$

LINEAR CODES

A linear code is a vectorial subspace of $\boldsymbol{F}_q^n$. It can be written like:

$$\boldsymbol{C} = \{\boldsymbol{x} = (x_1.x_2.....x_n) : x_i \in \boldsymbol{F}_q\}$$

These codes can also be defined by its parity check matrix and generator matrix. They satisfy that d(*c*) = w(*c*).

CYCLIC CODES

We say that a linear code is a cyclic code when:

$$c_0\, c_1 \dots c_{n-1} \in C \Rightarrow c_r \dots c_{n-1}\, c_0 \dots c_{r-1} \in C$$

An example could be:

$$c = \begin{Bmatrix} 0000000\ 1111111 \\ 1101000\ 0010111 \\ 0110100\ 1001011 \\ 0011010\ 1100101 \\ 0001101\ 1110010 \\ 1000110\ 0111001 \\ 0100011\ 1011100 \\ 1010001\ 0101110 \end{Bmatrix}$$

There are different ways to define this type of codes. In fact, we can use polynomials and zeros of polynomials. And there is a close relation between codes, ideals and divisors of x^n -1 as we can see behind.

Encoding and decoding are processes very important when we use codes. The key is that we have a message that would like to send then we rewrite the word and convert it in a codeword that is what we will send. There is not an only way to encode and decode. In each case we will have to contrast and choose best option.

BCH CODES

They are defined as:

$$\beta_q(n,\delta,w,b) = \{p(x) \in \tfrac{\boldsymbol{F}_q[x]}{x^n-1} : p(w^b) = \cdots = p(w^{b+\delta-2}) = 0\}$$

if w is an n-th root of unity.

For example:

δ Desing distance	Generator Polynomial	Dimension	Distance
1	1	7	1
3	x^3+x+1	4	3
5	$x^6+x^5+x^4+x^3+x^2+x+1$	1	7
7	$x^6+x^5+x^4+x^3+x^2+x+1$	1	7

REED-SOLOMON CODES

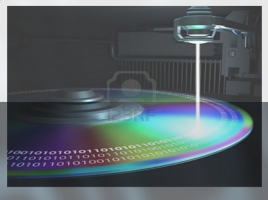
They appear from BCH codes making n = q-1, and they have the form:

$$R\delta_q(\delta,w,b,) = B_q(q-1,\delta,w,b)$$

This Kind of codes verify interesting properties and their applications are indubitable in everyday life. As we can see in the example that follows this lines.

APPLICATIONS CD

Actually, behind the usual CD´S there are two Reed-Solomon Codes which influence in encoding, and decoding, specially these codes are important because they allows detect, correct or even hide error that could exist because of stains and scratches on the cover of the CD.



Trabajo realizado por: **RAQUEL MARÍA HERNÁNDEZ FALCÓN**
Bajo la dirección de: **MARÍA VICTORIA REYES SÁNCHEZ**