Sección de Matemáticas
Universidad de La Laguna

Alejandro García García

Parámetros y decodificación eficiente de códigos afines

Parameters and efficient decoding of affine codes

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Junio de 2021

Dirigido por
Ignacio García Marco
Irene Márquez Corbella

Ignacio García Marco
Matemáticas, Estadística e I.O.
Universidad de La Laguna
38200 La Laguna, Tenerife

Irene Márquez Corbella
Matemáticas, Estadística e I.O.
Universidad de La Laguna
38200 La Laguna, Tenerife

Agradecimientos

A mi madre, por hacer todo lo posible y más para que pudiera seguir adelante y ayudarme a cumplir mi sueño.
A mis tutores Nacho e Irene, por su inmensa dedicación e implicación con este trabajo y por contagiarme la pasión y el amor que tienen por las matemáticas.

Alejandro García García
La Laguna, 9 de junio de 2021

Resumen · Abstract

Resumen

La Teoría de Códigos busca transmitir de manera eficiente y sin errores un mensaje entre un emisor y un receptor a través de un canal con ruido. Este ruido puede causar errores en el mensaje, luego el objetivo es recuperar el mensaje original a pesar de los errores que se hayan cometido. El emisor transformará el mensaje original mediante un proceso llamado codificación, añadiendo información redundante, y lo enviará por el canal. Una vez se reciba el mensaje, comienza el proceso más difícil, la decodificación, que consiste en recuperar el mensaje original a partir del mensaje recibido. Esta teoría busca familias de códigos que permitan corregir una cantidad considerable de errores y realizar los procesos de codificación y decodificación de manera eficaz. Una forma eficiente de realizar el proceso de codificación es utilizar aplicaciones lineales, lo que se traduce en usar, en particular, códigos lineales. En el capítulo dos introduciremos este tipo de códigos y sus propiedades.
Una familia interesante de códigos lineales son los códigos afines, en los que nos centraremos en el capítulo tres. En este capítulo uno de los objetivos será dar una cota para la distancia mínima de estos códigos. En la literatura esta cota es conocida, pero hace uso de herramientas algebraicas potentes como son las bases de Gröbner. Una de las novedades de este trabajo es presentar este resultado sin emplear esta herramienta. Al final del capítulo tres introduciremos las familias más conocidas de códigos afines. Para algunas de estas familias, presentaremos, en el capítulo cuatro, decodificadores eficientes.

Palabras clave: *Cuerpos finitos – Teoría de Códigos – Códigos lineales – Código afines*

Abstract

The main goal of Coding Theory is to efficient transfer reliable information through a channel with noise. This noise may distort the message so the aim is to recover the original message despite of the errors that may have occurred. The source will change the original message by a process called encoding, adding redundant information and, after that, the source will send the message through the channel. Once the encoded message is recieved, it starts the most difficult problem, decoding. The goal of this process is to obtain an estimate of the original message from the received message. This theory looks for families of codes that allow detecting and correcting as many errors as possible and that have efficient encoding and decoding procedures. An efficient way of encoding is to use linear maps, which is translated as using linear codes. This family of codes and its properties are introduced in Chapter two. An interesting family of linear codes are affine codes, we will focus our attention to these codes in Chapter three. One of the main goals of this Chapter is to give an lower bound of the minimum distance. In the literature this bound is known, but it makes use of powerful algebraic tools such as Gröbner bases. One of the key ideas of this Chapter is to present this result but free of Gröbner tools. At the end of Chapter three we will introduce some well known families of affine codes. For some of these families we will present, in Chapter four, efficient decoding algorithms.

Keywords: *Finite fields – Coding Theory – Linear codes – Affine codes*

Contenido

Introducción

En el año 1948 Claude Shannon publicó un artículo en dos partes, en los números de julio y octubre de Bell System Technical Journal, llamado "*A mathematical theory of communication*", que significó el comienzo de la Teoría de Códigos y de Teoría de la Información. Dado un canal de comunicación que puede alterar la información que se pretende transmitir a través de él, Shannon definió la *capacidad del canal* y demostró que se podía comunicar información de manera fiable siempre que no se supere su capacidad. Sin embargo la prueba no es constructiva, luego no hay una manera específica de enviar mensajes con alta fiabilidad, aunque muestra que es posible.

El objetivo de la Teoría de Códigos es transferir de manera eficiente y sin errores un mensaje entre un emisor y un receptor. Los mensajes pueden ser dañados durante la transmisión y el objetivo es recuperar el mensaje original a pesar de los errores que se hayan cometido. El proceso de comunicación con códigos correctores tiene diferentes partes. Primero elegimos un alfabeto $\mathcal{A}$, a lo largo de este trabajo consideraremos que nuestro alfabeto será $\mathcal{A} = \mathbb{F}_q$, el cuerpo finito con q elementos. Los mensajes se representan como una k-upla de elementos de $\mathcal{A}$, es decir $m = (m_1, \ldots, m_k) \in \mathcal{A}^k = \mathbb{F}_q^k$. Luego, transformamos el mensaje utilizando un proceso de codificación, añadiendo información redundante. Es decir, transformamos cada mensaje $m \in \mathbb{F}_q^k$ en una palabra de un código $\mathcal{C}$ (subconjunto no vacío de $\mathbb{F}_q^n$) con $k < n$ utilizando una aplicación biyectiva que definimos como *codificación*:

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathcal{C} \subseteq \mathbb{F}_q^n \\ m &\longmapsto Enc(m) = c \end{aligned}$$

Tras esto, el mensaje se envía a través de un *canal* que lo puede perturbar, un canal es cualquier medio de transmisión, como puede ser el cable de teléfono, la fibra óptica, las ondas de radio, los CD, los móviles y, en general, cualquier dispositivo de comunicación electrónica o de almacenamiento magnético. Nosotros siempre vamos a trabajar con ciertas reglas. Vamos a suponer que el canal

utilizado es discreto y sin memoria. Es decir, que el conjunto de símbolos de entrada y de salida son finitos y que la transmisión de un símbolo no está influenciada por la transmisión de los símbolos anteriores, por lo tanto transmitir un símbolo a través del canal significa asociar a dicho símbolo, aleatoriamente, un símbolo según una distribución de probabilidades asociada al canal.

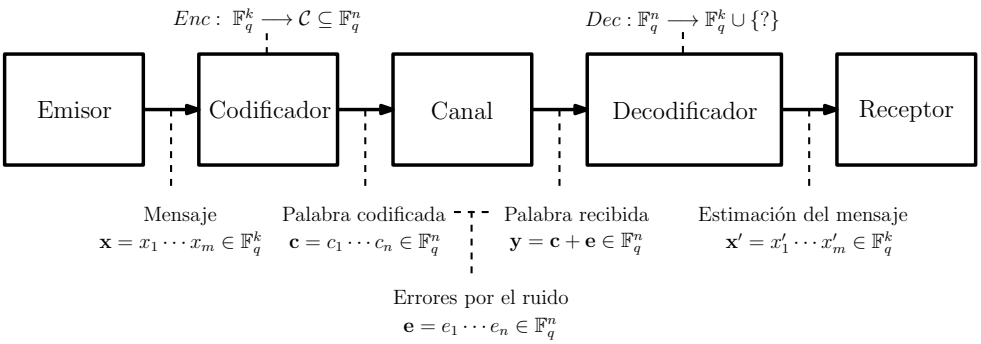


Figura 0.1: Ejemplo de comunicación con códigos correctores.

Finalmente, el proceso de decodificación, en el que se busca recuperar el mensaje original $m \in \mathbb{F}_q^k$. El mensaje recibido será $y = c + e \in \mathbb{F}_q^n$, donde $c \in \mathcal{C}$ es la palabra enviada y $e \in \mathbb{F}_q^n$ es el error que se ha producido en la transmisión. Un ejemplo de decodificador por mínima distancia que corrige s errores es una aplicación: $Dec : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k \cup \{?\}$, tal que si $Enc : \mathbb{F}_q^k \longrightarrow \mathcal{C} \subseteq \mathbb{F}_q^n$, es un codificador para el código $\mathcal{C}$, entonces $Dec(Enc(m + e)) = m$, para todo $e \in \mathbb{F}_q^n$ con $w_H(e) \leq s$.

La Teoría de Códigos busca familias de códigos que permitan realizar los procesos de codificación y decodificación de forma rápida y permita corregir un gran número de errores. La decodificación eficiente es una tarea difícil, por lo que es una línea activa de investigación.

El matemático Richard Hamming en 1940, fue pionero en Teoría de Códigos, llegando a inventar en 1950 el primer código corrector, el código de Hamming. Este investigador además de los códigos de Hamming, inventó los conceptos de ventanas de Hamming, números de Hamming y distancia de Hamming, en particular esta última será fundamental para el desarrollo de esta área.

Un ejemplo claro de comunicación con códigos correctores es la comunicación entre la ISS, Estación Espacial Internacional, y una estación receptora de la Tierra. Pero en la vida cotidiana también utilizamos los códigos correctores de forma frecuente. Por ejemplo, para obtener la letra del DNI, se toma como alfabeto las letras del abecedario español excluyendo las letras O, I, U y Ñ evitando así confusiones entre los números 0 y 1 y las letras O e I, y entre las letras V y N y las letras U y Ñ. Obtenemos así un alfabeto de tamaño 23. A cada letra del alfabeto le asignamos un número del 1 al 23 de la siguiente manera:

Número	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Letra	A	B	C	D	E	F	G	H	J	K	L	M	N	P	Q	R	S	T	V	W	X	Y	Z

Tabla 0.1: Asignación de letras del DNI.

La codificación en este caso es el resultado de añadirle a un número x de 8 dígitos una letra. Esta letra se obtiene de calcular el resto al dividir el número x entre 23 y buscar la letra asignada correspondiente a ese resto en la Tabla. Es decir, el número del DNI y el número asociado a nuestra letra del DNI son congruentes módulo 23.

Otro ejemplo que utilizamos en la vida cotidiana es el ISBN. Al principio, el ISBN estaba compuesto por 10 dígitos. Sin embargo, estos códigos resultaron escasos y en el año 2007 se decidió implantar un nuevo código ISBN de 13 dígitos compatible con el anterior. Desde entonces el ISBN está formado por 5 grupos de dígitos en lugar de 4, y va precedido por el 978, que identifica al producto libro. También ha cambiado la forma de calcular el último número del ISBN, el dígito de control.

Identificación de libro	País o lengua	Editorial	Número de ejemplar	Dígito de control
978	Hasta 5 dígitos	Hasta 6 dígitos	Hasta 6 dígitos	Dígito del 0 al 9

Tabla 0.2: Estructura del ISBN

El dígito de control de un ISBN-13 utiliza un algoritmo basado en el módulo 10. Se multiplica el primero de los 12 números iniciales por 1, el segundo por 3, el tercero por 1, el cuarto por 3, y así sucesivamente hasta llegar al número 12. El dígito de control es el valor que se debe añadir a la suma de todos estos productos para hacerla divisible por 10. Si el resultado de la suma ya fuese múltiplo de 10, el dígito de control sería 0.

Resumen de esta memoria

Esta memoria está dividida en cuatro partes. Un primer capítulo en el que describiremos conceptos básicos de cuerpos finitos, el alfabeto que vamos a utilizar en nuestra comunicación con códigos. Uno de los resultados principales del capítulo será la existencia de un polinomio mónico irreducible de grado determinado cuyos coeficientes pertenecen a un cuerpo finito, que nos permitirá asegurar la construcción de algunos cuerpos finitos.

En el segundo capítulo introduciremos los conceptos básicos de la Teoría de Códigos como la distancia de Hamming, la decodificación y la capacidad correctora de un código. Una forma eficiente de realizar el proceso de codificación es utilizar aplicaciones lineales, lo que se traduce en usar códigos lineales, que son subespacios de $\mathbb{F}_q^n$. Los códigos afines son una subfamilia de códigos lineales.

El tercer capítulo estará dedicado a los códigos de evaluación o códigos afines. Partiremos de un resultado que nos acotará el número de soluciones de un sistema de ecuaciones, que será de utilidad para poder acotar la distancia mínima de este tipo de códigos mediante la función huella. En la literatura esta cota es conocida, pero hace uso de herramientas algebraicas potentes como son las bases de Gröbner. Una de las novedades de este trabajo es presentar este resultado sin emplear esta herramienta. Otro de lo resultados de este capítulo es presentar las familias de códigos afines notables, como son los códigos Reed-Solomon, los códigos Reed-Muller, los códigos Cubo y los códigos Hiperbólicos, y definir sus parámetros.

Por último en el cuarto capítulo hablaremos de decodificación de códigos afines. En particular, se presentarán algoritmos de decodificación conocidos y eficientes para algunas de las familias de códigos más importantes: los códigos Reed-Solomon, los códigos Reed-Muller binarios y los códigos Cubo.

En esta memoria se intentará plasmar con todo detalle los resultados y las pruebas con el objetivo de que sea una memoria autocontenida. Es por ello que al construir familias de códigos, presentar algoritmos o exponer resultados algebraicos se incluirán ejemplos y demostraciones de los casos más sencillos.

1

Cuerpos finitos

En este capítulo hablaremos sobre cuerpo finitos y algunas de sus propiedades, ya que estos serán el alfabeto finito que utilizaremos en la comunicación con códigos correctores en los siguientes capítulos. Para construir cuerpos finitos con $q = p^r$ elementos, en la práctica se considera el anillo cociente entre un cuerpo finito $\mathbb{Z}_p$ con p elementos y el ideal generado por $p(x) \in \mathbb{Z}_p[x]$ un polinomio irreducible de grado r. En este capítulo justificaremos la existencia de este polinomio y por tanto, esta construcción.

1.1. Cuerpos finitos

Dado $\mathbb{K}$ un conjunto, con dos operaciones cerradas, una aditiva $+$ y una multiplicativa $\cdot$, se dice que $(\mathbb{K}, +, \cdot)$ es un cuerpo si:

1. $(\mathbb{K}, +)$ es un grupo abeliano con elemento neutro $0 \in \mathbb{K}$.
2. $(\mathbb{K} - \{0\}, \cdot)$ es un grupo abeliano.
3. Se satisface la siguiente ley distributiva: $(a+b)\cdot c = (a\cdot c)+(b\cdot c)$, $\forall a, b, c \in \mathbb{K}$.

Equivalentemente, $\mathbb{K}$ es cuerpo si $(\mathbb{K}, +, \cdot)$ es un anillo conmutativo y unitario en el que cualquier elemento distinto del elemento neutro para la operación aditiva tiene inverso.

Cuando $(\mathbb{K}, +, \cdot)$ es un cuerpo finito, al número de elementos de $\mathbb{K}$ se le denomina orden de $\mathbb{K}$. Se denota por $\mathbb{F}_q$ al cuerpo finito con q elementos. Veremos en el Teorema 1.4 que este cuerpo es único salvo isomorfismos.

Sea A un anillo unitario con 1_A el neutro para el producto, el homomorfismo característico de A es el único homomorfismo de anillos de $\mathbb{Z}$ en A, que se define de la siguiente manera:

$$\begin{aligned} \varphi_A : \mathbb{Z} &\longrightarrow A \\ 1 &\longmapsto 1_A \end{aligned}$$

$$n(>0) \longmapsto 1_A + \overset{n \text{ veces}}{\cdots} + 1_A$$

Dado que $\mathbb{Z}$ es un dominio de ideales principales, existe $b \in \mathbb{N}$ tal que el núcleo de la aplicación será $\mathrm{Ker}(\varphi_A) = (b)\mathbb{Z} = (b), b \in \mathbb{Z}^+$. A este entero b se le llama la *característica* de A y se denota por $\mathrm{car}(A)$. Por otro lado, si b es no nulo, entonces $b = \mathrm{mín}\{\ell \in \mathbb{Z}^+ \mid 1_A + \overset{\ell \text{ veces}}{\cdots} + 1_A = 0\}$.

Proposición 1.1. *Sea A un dominio de integridad, entonces* $\mathrm{car}(A) = 0$ *ó* $\mathrm{car}(A) = p$*, con p número primo.*

Demostración. Sea $b = \mathrm{car}(A)$ y supongamos que b es distinto de cero. Veamos que es primo, procedemos por reducción al absurdo. Es decir, supongamos que $b = rs$ con $1 < r, s < b$, se tiene que $0_A = \varphi_A(m) = \varphi_A(r)\varphi_A(s)$. Como A es un dominio de integridad, entonces $\varphi_A(r) = 0$ ó $\varphi_A(s) = 0$ llegando a una contradicción ya que b era el menor entero no nulo cuya imagen era nula. □

Si $\mathbb{F}_q$ es cuerpo finito, entonces la aplicación característica no puede ser inyectiva y, por la Proposición 1.1, se puede afirmar que $\mathrm{car}(\mathbb{F}_q)$ es un número primo.

Corolario 1.2. *Sea $\mathbb{F}_q$ un cuerpo finito con característica p, entonces $\mathbb{F}_q$ contiene un subcuerpo L isomorfo a $\mathbb{Z}_p$. Además L es el subcuerpo más pequeño de $\mathbb{F}_q$.*

Demostración. Aplicando el Primer Teorema de Isomorfía al homomorfismo característico, $\mathbb{Z}_p \cong \mathrm{Im}(\varphi_{\mathbb{F}_q}) \subseteq \mathbb{F}_q$. Además es el más pequeño, ya que de estar $1_{\mathbb{F}_q}$ entonces deberá estar $n \cdot 1_{\mathbb{F}_q}$, para todo $n \in \mathbb{Z}$, en particular, estarán todos los elementos de L. □

1.1.1. Construcción de cuerpos finitos

En lo que sigue, p denota un número primo. Antes de hablar de la construcción de cuerpos finitos, veamos unos teoremas que nos serán de utilidad posteriormente.

Teorema 1.3. *Sea $\mathbb{F}_q$ un cuerpo finito y sea p la característica de $\mathbb{F}_q$, entonces se tiene:*

1. *$\mathbb{F}_q$ contiene un subcuerpo $\mathbb{F}_p$ de p elementos.*
2. *$\mathbb{F}_q$ es un $\mathbb{F}_p$-espacio vectorial.*
3. *$q = p^r$, para algún p primo y $r \in \mathbb{Z}^+$.*
4. *$p \cdot \alpha = 0$, $\forall \alpha \in \mathbb{F}_q$.*
5. *$(x + y)^{p^s} = x^{p^s} + y^{p^s}$, $\forall x, y \in \mathbb{F}_q$, para todo $s \in \mathbb{Z}^+$.*
6. *$x^q = x$, $\forall x \in \mathbb{F}_q$.*

Demostración. 1. Por el Corolario 1.2.

2. Veamos que $\mathbb{F}_q$ tiene estructura de $\mathbb{F}_p$-espacio vectorial. Por definición de cuerpo sabemos que $(\mathbb{F}_q\ ,+)$ es grupo abeliano y por el apartado 1. se tiene que $\mathbb{F}_p \subseteq \mathbb{F}_q$, por lo tanto, podemos definir un producto por restricción de escalares: $\forall \lambda \in \mathbb{F}_p, \forall a \in \mathbb{F}_q, \lambda \cdot a \in \mathbb{F}_q$. Además como $\mathbb{F}_q$ es finito, se tiene que $\dim_{\mathbb{F}_p}(\mathbb{F}_q) < \infty$.
3. Acabamos de ver en el apartado 2. que $\mathbb{F}_q$ es un $\mathbb{F}_p$-espacio vectorial de dimensión $r = \dim_{\mathbb{F}_p}(\mathbb{F}_q) < \infty$. Sea $\{\alpha_1, \ldots, \alpha_r\}$ una base de $\mathbb{F}_q$ sobre $\mathbb{F}_p$. Entonces, $\forall \alpha \in \mathbb{F}_q$ existen unos únicos $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_p$ tales que $\alpha = \lambda_1\alpha_1 + \cdots + \lambda_r\alpha_r$ con $\lambda_i \in \mathbb{F}_q$. De donde se deduce que $|\mathbb{F}_q| = q = p^r$.
4. Como la característica de $\mathbb{F}_q$ es p, se tiene que $\forall \alpha \in \mathbb{F}_q$:

$$p \cdot \alpha = (\overbrace{\alpha + \cdots + \alpha}^{p \text{ veces}}) = \alpha\ (\overbrace{1 + \cdots + 1}^{p \text{ veces}}) = \alpha \cdot 0 = 0.$$

5. Sea $q = p^r$. Procederemos por inducción sobre r. Para $r = 1$:

$$(x+y)^p = \binom{p}{0} x^p + \binom{p}{1} x^{p-1}y + \cdots + \binom{p}{p-1} xy^{p-1} + \binom{p}{p} y^p.$$

   Como p es primo, entonces se tiene que $\binom{p}{i}$ es divisible por p para todo $i = \{1, \ldots, p-1\}$, se deduce del apartado 4. que $\binom{p}{i}$ se anulan. Por tanto, $(x+y)^p = x^p + y^p$.
   Supongamos que el enunciado se cumple para $s < r$, comprobemos que se cumple para r:

$$(x+y)^{p^r} = ((x+y)^{p^{r-1}})^p = (x^{p^{r-1}} + y^{p^{r-1}})^p = x^{p^r} + y^{p^r} = x^q + y^q.$$

6. Si $\alpha = 0$, entonces $\alpha^q = 0$. Ahora, consideramos $\alpha \in \mathbb{F}_q \backslash \{0\}$. Nótese que $\mathbb{F}_q^* = \mathbb{F}_q \backslash \{0\}$ es un grupo multiplicativo de orden $q-1$. Luego, como el orden de un elemento divide al orden del grupo (Teorema de Lagrange para grupos), tenemos que $\alpha^{q-1} = 1$, y por lo tanto $\alpha^q = \alpha$.

□

Esta última demostración nos ha aportado que los ceros de $p(x) = x^q - x$ son precisamente los elementos de $\mathbb{F}_q$, o equivalentemente, que $p(x)$ se descompone en factores lineales en $\mathbb{F}_q$.

Teorema 1.4. *Para todo p primo y todo $r \in \mathbb{Z}^+$, existe un cuerpo finito con $q = p^r$ elementos. Además este cuerpo es único salvo isomorfismo.*

Demostración. Definimos $F(x) = x^q - x \in \mathbb{F}_p[x]$. Se puede ver que su primera derivada $F'(x) = q \cdot x^{q-1} - 1 = 1$ en $\mathbb{F}_p[x]$, como $F(x)$ y $F'(x)$ son coprimos, sabemos que $F(x)$ no tiene raíces múltiples. Sean L el cuerpo de descomposición

de $F(x)$ sobre $\mathbb{F}_q$ y $R = \{\alpha_1, \ldots, \alpha_q\}$ el conjunto de raíces de $F(x)$ en la clausura algebraica de $\mathbb{F}_p$ veamos que R con la operación aditiva y multiplicativa es un cuerpo, para ello, veamos que es un subanillo de L y además que todo elemento tiene inverso.
Sean $\alpha_1, \alpha_2 \in R$:

1. Por Teorema 1.3(5) tenemos que $(\alpha_1 - \alpha_2)^q = \alpha_1^q + (-1)^q \alpha_2^q =$

$$= \begin{cases} \alpha_1^q - \alpha_2^q = \alpha_1 - \alpha_2, \ \textit{si } q \textit{ es impar.} \\ \alpha_1^q + \alpha_2^q = \alpha_1 + \alpha_2 = \alpha_1 - \alpha_2, \ \textit{si } q \textit{ es par.} \end{cases}$$

   Este último caso ocurre cuando $\text{car}(A) = 2$.
2. Por el Teorema 1.3(6), $(\alpha_1 \alpha_2)^q = \alpha_1^q \alpha_2^q = \alpha_1 \alpha_2$.
3. $(1_L)^q = 1_L$, por tanto $1_L \in R$.
4. Si $\alpha \in R$, $\alpha \neq 0$, entonces $(\alpha_1^{-1})^q = (\alpha_1^q)^{-1} = \alpha_1^{-1}$, por tanto $\alpha^{-1} \in R$.

Sea $\mathbb{K}$ otro cuerpo con q elementos, sabemos por el Teorema 1.3 que los q elementos de $\mathbb{K}$ son las raíces de $x^q - x$, luego podemos identificar cada elemento de $\mathbb{K}$ con uno de R, obteniendo el resultado. □

En la práctica, para construir el cuerpo finito $\mathbb{F}_q$ con $q = p^r$ elementos se siguen los siguientes pasos:

1. Consideramos el cuerpo finito con p elementos: $\mathbb{F}_p \cong \mathbb{Z}_p$.
2. Buscamos un polinomio irreducible $p(x)$ de grado r con coeficientes en $\mathbb{F}_p$.
3. Consideramos el anillo cociente entre $\mathbb{F}_p[x]$ y el ideal generado por $p(x)$, es decir $\mathbb{F}_p/(p(x))$.

Como $p(x)$ es irreducible, $\mathbb{F}_p$ es dominio de ideales principales, el ideal $(p(x))$ es maximal, lo que implicaría que $\mathbb{F}_p/(p(x))$ es un cuerpo finito con p^r elementos. El punto clave en esta construcción es la existencia de $p(x) \in \mathbb{F}_p[x]$ un polinomio irreducible de grado r, dedicamos la siguiente subsección a justificar su existencia.

1.1.2. Existencia de un polinomio irreducible de grado r en $\mathbb{F}_q$

En esta sección no solamente se justificará la existencia de polinomios irreducibles de grado r en $\mathbb{F}_q$ sino que se contarán cuántos polinomios existen con esta propiedad. Más concretamente, vamos a demostrar la siguiente fórmula atribuida a Gauss.

Teorema 1.5. *El número de polinomios mónicos irreducibles de grado r que existen sobre un cuerpo finito $\mathbb{F}_q$ viene dado por:*

$$\frac{1}{r} \sum_{d|r} \mu\left(\frac{r}{d}\right) q^d, \tag{1.1}$$

donde μ es la función de Möbius y d son los divisores positivos de r.

La suma se efectúa sobre los divisores d de r, incluyendo al propio r y $\mu(x)$ es la función aritmética de Möbius. Si x es un número entero positivo y $p_1^{\alpha_1} \cdots p_s^{\alpha_s}$ su descomposición en factores primos, se tiene que la función aritmética de Möbius se define como:

$$\mu(x) = \begin{cases} 0, & \textit{si existe } \alpha_i \geq 2. \\ (-1)^s, & \textit{si } \alpha_i = 1 \textit{ para todo } i. \end{cases}$$

En esta memoria presentaremos una prueba de esta fórmula extraída de [4], antes de probar el resultado introducimos algunos resultados previos.

Lema 1.6.

1. *Las raíces de un polinomio irreducible sobre $\mathbb{F}_q$ son siempre distintas.*
2. *Dos polinomios mónicos irreducibles distintos sobre $\mathbb{F}_q$ no pueden tener una raíz en común.*

Demostración. 1. Sea $p(x)$ irreducible sobre $\mathbb{F}_q$ y sea $\mathbb{F}$ su cuerpo de descomposición. Entonces por el Teorema 1.3(6), $\mathbb{F}$ es un cuerpo finito y existe un entero $m > 0$ tal que $\mathbb{F} = \{\alpha_1, \ldots, \alpha_{q^m}\}$ donde $\alpha_1, \ldots, \alpha_{q^m}$ son las raíces del polinomio $x^{q^m} - x$ luego $p(x)$ divide a $x^{q^m} - x$ y además α_i son distintos, con $i = 1, \ldots, q^m$.

2. Supongamos que existen $p(x), q(x) \in \mathbb{F}_q[x]$ irreducibles, con una raíz en común y $p(x) \neq q(x)$. Por la definición y unicidad del polinomio mínimo, si α es esa raíz en común, se tendría que $p(x) = m_{\alpha,\mathbb{F}_q}(x) = q(x)$, lo que contradice nuestra hipótesis. □

Lema 1.7. *$\mathbb{F}_{q^\alpha} \subseteq \mathbb{F}_{q^\beta}$ si y solo si α divide a β.*

Demostración. Supongamos que $\mathbb{F}_{q^\alpha} \subseteq \mathbb{F}_{q^\beta}$, sabemos que $q = p^r$, luego tenemos la siguiente torre de cuerpos: $\mathbb{F}_{q^\alpha} \hookrightarrow \mathbb{F}_{q^\beta}$. Luego, como $\mathbb{F}_{q^\beta}$ es un $\mathbb{F}_{q^\alpha}$ espacio vectorial de dimensión d, con $[\mathbb{F}_{q^\beta} : \mathbb{F}_{q^\alpha}] = d$. Entonces:

$$q^\beta = |\mathbb{F}_{q^\beta}| = |(\mathbb{F}_{q^\alpha})^d| = (q^\alpha)^d.$$

Por tanto, $\beta = \alpha d$ (α divide a β).

Recíprocamente, si α divide a β se tiene que $\beta = \alpha \cdot s$, con $s \in \mathbb{Z}$. Por el Teorema 1.3(6). sabemos que $\mathbb{F}_{q^\alpha}$ son las raíces de $p(x) = x^{q^\alpha} - x$. Y por otro lado, sabemos que $\mathbb{F}_{q^\beta}$ son las raíces de $q(x) = x^{q^\beta} - x = x^{q^{\alpha s}} - x$. De aquí se concluye que $\mathbb{F}_{q^\alpha} \subseteq \mathbb{F}_{q^\beta}$. □

Definición 1.8. *Sea $m \in \mathbb{Z}^+$ y $L \subsetneq \mathbb{F}_{q^m}$, decimos que L es un subcuerpo maximal si no hay ningún cuerpo intermedio entre L y $\mathbb{F}_{q^m}$.*

Proposición 1.9. *Sea L un subcuerpo de $\mathbb{F}_{q^m}$. Entonces L es un subcuerpo maximal de $\mathbb{F}_q^m$ si y solo si $L = \mathbb{F}_{q^s}$ y $\frac{m}{s}$ es un número primo.*

Demostración. Como $L \subsetneq \mathbb{F}_{q^m}$, entonces $L = \mathbb{F}_{q^s}$ con s divisor de m por el Lema 1.7. Sea $a \in \mathbb{Z}^+$, $a \geq 2$, tal que $m = as$, veamos que L es maximal si y solo si a es primo.

Supongamos que a es primo y sea K cuerpo tal que $L \subseteq K \subseteq \mathbb{F}_{q^m}$, entonces $K = \mathbb{F}_{q^r}$ y $s \mid r$, $r \mid m$, luego $r \mid a$, por tanto $r = 1$ ó $r = a$. Entonces, $K = L$ o $K = \mathbb{F}_{q^m}$. Si a no es primo, tomando t divisor primo de a tenemos que:

$$L = \mathbb{F}_{q^s} \subsetneq \mathbb{F}_{q^{ts}} \subsetneq \mathbb{F}_{q^{as}} = \mathbb{F}_{q^m}.$$

Concluyendo así que L no es maximal. □

Proposición 1.10. $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} = \mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}}$.

Demostración. Sabemos que la intersección de cuerpos es un cuerpo. Veamos que $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} \subseteq \mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}}$. En efecto, se tiene que:

$$\alpha = [\mathbb{F}_{q^\alpha} : \mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta}][\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} : \mathbb{F}_q].$$

$$\beta = [\mathbb{F}_{q^\beta} : \mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta}][\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} : \mathbb{F}_q].$$

Luego $[\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} : \mathbb{F}_q]$ divide a α y además $[\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} : \mathbb{F}_q]$ divide a β. Por tanto, $[\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} : \mathbb{F}_q]$ divide a $\mathrm{mcd}(\alpha, \beta)$, entonces por Lema 1.7 se tiene que $\mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta} \subseteq \mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}}$.

Recíprocamente, $\mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\alpha}$ y $\mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\beta}$, entonces:

$$\mathbb{F}_{q^{\mathrm{mcd}(\alpha,\beta)}} \subseteq \mathbb{F}_{q^\alpha} \cap \mathbb{F}_{q^\beta}.$$

□

Proposición 1.11. *Sea α un elemento algebraico sobre $\mathbb{F}_q$, los siguientes enunciados son equivalentes:*

1. $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = r$.
2. *α no pertenece a ningún subcuerpo propio de $\mathbb{F}_{q^r}$.*
3. *α no pertenece a ningún subcuerpo maximal de $\mathbb{F}_{q^r}$.*

Demostración. Dado que $\mathbb{F}_q(\alpha)$ es el menor cuerpo que contiene a $\mathbb{F}_q$ y a α, se tiene que $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = r$ si y solo si $\mathbb{F}_q(\alpha) = \mathbb{F}_{q^r}$ y de aquí 1. es equivalente a 2.

Dado que todo subcuerpo maximal es propio, se tiene que 2. implica 3. Además, como todo subcuerpo propio está contenido en un maximal, se tiene la equivalencia.

□

Ahora ya podemos proceder con la demostración del Teorema 1.5.

Demostración del Teorema 1.5.

Si $r = 1$, existen q polinomios irreducibles y mónicos de grado 1, ya que $\mathbb{F}_q$ tiene q elementos, además, con $r = 1$, q es el valor que nos devuelve la fórmula (1.1). De aquí en adelante supondremos que $r > 1$. Sean:

$$P_r = \{p(x) : p(x) \text{ mónico de grado } r \text{ e irreducible en } \mathbb{F}_q[x]\}. \tag{1.2}$$

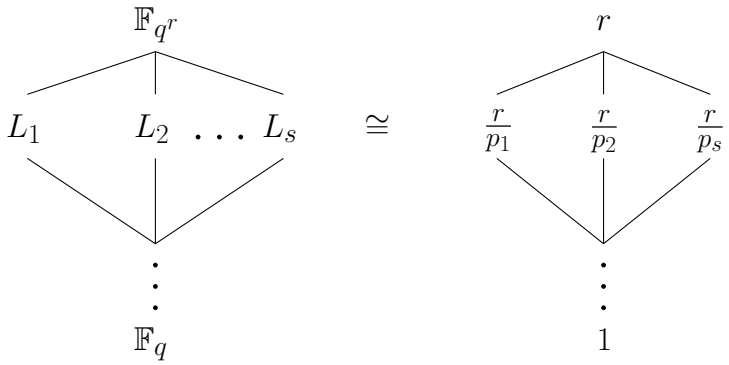
$$R_r = \bigcup_{p(x)\in P_r} \{\alpha/\alpha \text{ raíz de } p(x)\}. \tag{1.3}$$

Es fácil comprobar que $R_r \subseteq \mathbb{F}_{q^r}$, además por el Lema 1.6, cada polinomio aporta r elementos diferentes a R_r, concluyendo así que $|R_r| = r \cdot |P_r|$. Ahora bien, en virtud de la Proposición 1.11:

$$\begin{aligned} R_r &= \{\alpha \in \mathbb{F}_{q^r} \mid [\mathbb{F}_q(\alpha) : \mathbb{F}_q] = r\} \\ &= \{\alpha \in \mathbb{F}_{q^r} \mid \alpha \notin L \ , \ L \subsetneq \mathbb{F}_{q^r}\} \\ &= \{\alpha \in \mathbb{F}_{q^r} \mid \alpha \notin L \ , \ L \subsetneq \mathbb{F}_{q^r} \ , \ L \text{ maximal}\}. \end{aligned}$$

Sea $r = p_1^{\alpha_1} \cdots p_s^{\alpha_s}$ su descomposición en factores primos, entonces, por la Proposición 1.9 los subcuerpos propios maximales son de la forma $L_1 = \mathbb{F}_{q^{\frac{r}{p_1}}}$, $L_2 = \mathbb{F}_{q^{\frac{r}{p_2}}}$, $\ldots$, $L_s = \mathbb{F}_{q^{\frac{r}{p_s}}}$.

De aquí, podemos afirmar, por la tercera igualdad de la Proposición 1.11 que $|R_r| = |(L_1 \cup \ldots \cup L_S)^c|$. Además, por el Lema 1.7 vemos que el poset[1] correspondiente a los subcuerpos de $\mathbb{F}_{q^r}$ ordenados por inclusión es isomorfo al poset correspondiente a los divisores de r, con el isomorfismo definido por la relación $\mathbb{F}_{q^t} \mapsto t$.



Por último, por la Proposición 1.10 podemos afirmar que para todo $I \subseteq \{1, \ldots, s\}$ tenemos que $\cap_{i\in I} L_i = \mathbb{F}_{q^s}$, siendo $s = \frac{r}{\prod_{i\in I} p_i}$:

$$L_1 \cap L_2 = \mathbb{F}_{q^{\frac{r}{p_1 \cdot p_2}}}, L_1 \cap L_2 \cap L_3 = \mathbb{F}_{q^{\frac{r}{p_1 \cdot p_2 \cdot p_3}}} \ , \text{ etc.}$$

[1] En Teoría del Orden, un poset es un conjunto equipado con una relación binaria de orden parcial. En este caso, la relación de orden parcial es la inclusión y el conjunto elegido es el formado por los subcuerpos de $\mathbb{F}_{q^r}$

Por tanto, usando la definición de P_r (1.2) y el principio de Inclusión-Exclusión [2] , tenemos que:

$$|L_1 \cup \ldots \cup L_S| = |L_1| + |L_2| + \cdots + |L_s| - |\mathbb{F}_{q^{\frac{r}{p_1 p_2}}}| + \cdots + (-1)^r |\mathbb{F}_{q^{\frac{r}{p_1 \cdots p_r}}}|.$$

Entonces como $|R_r| = |(L_1 \cup \ldots \cup L_S)^c|$, se tiene que:

$$\begin{aligned} |R_r| &= |\mathbb{F}_{q^r}| - |L_1| - |L_2| - \cdots - |L_s| + |\mathbb{F}_{q^{\frac{r}{p_1 p_2}}}| + \cdots + (-1)^r |\mathbb{F}_{q^{\frac{r}{p_1 \cdots p_r}}}| \\ &= q^r - q^{\frac{r}{p_1}} - q^{\frac{r}{p_2}} - \cdots - q^{\frac{r}{p_r}} + q^{\frac{r}{p_1 p_2}} + \cdots + (-1)^r q^{\frac{r}{p_1 \cdots p_r}} \\ &= \frac{1}{r} \sum_{d|r} \mu\left(\frac{r}{d}\right) q^d. \end{aligned}$$

□

Para finalizar, hemos contado los polinomios mónicos, pero si queremos contar los polinomios no mónicos en $\mathbb{F}_q$, bastará multiplicar $|P_r|$ por $q-1$, que son todos los elementos posibles no nulos de $\mathbb{F}_q$.

Corolario 1.12. *Para todo q número primo y para cualquier $r \in \mathbb{Z}^+$, existe un polinomio $p(x)$ con coeficientes en $\mathbb{F}_q$ irreducible de grado r.*

Demostración.

$$\begin{aligned} &\frac{1}{r} \sum_{d|r} \mu\left(\frac{r}{d}\right) q^d \geq \frac{1}{r}(q^r - \sum_{i<r} \delta_i q^i) \geq \frac{1}{r}(q^r - q^{r-1} - \cdots - 1) = \\ &= \frac{q^r}{r}(1 - \frac{1}{q} - \cdots - \frac{1}{q^r}) = \frac{q^r}{r}(1 - \sum_{1=i}^{r} \frac{1}{q^i}) > \\ &> \frac{q^r}{r}(1 - \sum_{i=1}^{\infty} \frac{1}{q^i}) = \frac{q^r}{r}(1 - \frac{1}{q-1}) \geq 0. \end{aligned}$$

De aquí extraemos que:

$$\frac{1}{r} \sum_{d|r} \mu\left(\frac{r}{d}\right) q^d > 0.$$

□

[2] El Principio de Inclusión-Exclusión: Sean $A_1, \ldots, A_s$ conjuntos, entonces se tiene que: $|\cup_{i=1}^{s} A_i| = \sum_{\substack{I \subseteq \{1,\ldots,s\} \\ I \neq \emptyset}} (-1)^{|I|+1} |\cap_{i \in I} A_i|$, esta expresión recibe el nombre de Principio de Inclusión-Exclusión.

2

Teoría de Códigos

Un código es un subconjunto $\mathcal{C} \subseteq \mathbb{F}_q^n$. Si $\mathcal{C}$ tiene M elementos, se dice que $\mathcal{C}$ es un $(n, M)_q$-código. Cada mensaje $m \in \mathbb{F}_q^k$, con $k < n$ lo transformamos en una palabra del código $\mathcal{C}$, utilizando una aplicación biyectiva:

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathcal{C} \subseteq \mathbb{F}_q^n \\ m &\longmapsto Enc(m) = c \end{aligned}$$

A este proceso lo llamamos *codificación*. Por otro lado, el proceso de decodificación buscará recuperar el mensaje original m a partir de la palabra de código recibido. Este proceso será más delicado debido a que el canal por el que se ha enviado el mensaje puede producir errores en el mismo. En particular, nos centraremos en los códigos lineales que nos permitirán usar herramientas del álgebra lineal.

2.1. Distancia y peso de Hamming

Definición 2.1. *Para dos vectores $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ se llama distancia de Hamming entre x e y al número de posiciones en las que el vector x y el vector y difieren y se denota por $d_H(x, y)$.*

$$d_H(x, y) = |\{i \mid x_i \neq y_i\}| \ , \ \forall x, y \in \mathbb{F}_q^n.$$

Proposición 2.2. *$d_H(x, y)$ es una métrica, es decir, $d_H(x, y)$ verifica:*

1. *$d_H(x, y) \geq 0$ y $d_H(x, y) = 0$ si y solo si $x = y$.*
2. *$d_H(x, y) = d_H(y, x)$.*
3. *$d_H(x, y) \leq d_H(x, z) + d_H(z, y)$ (desigualdad triangular).*

Demostración. Las propiedades 1. y 2. se tienen por definición. Para probar 3. definimos, para todo $x, y \in \mathbb{F}_q^n$, $D(x, y) = \{i \mid x_i \neq y_i\}$ entonces se tiene que $|D(x, y)| = d_H(x, y)$. Luego, consideremos para todo $z \in \mathbb{F}_q^n$:

$$D(x,y)^c = \{i \mid x_i = y_i\} = \{i \mid x_i = y_i = z_i\} \cup \{i \mid x_i = y_i \neq z_i\} \subseteq \{1, \ldots, n\},$$

$$|D(x,y)^c| \geq |\{i \mid x_i = y_i = z_i\}| = |\{i \mid x_i = z_i\} \cap \{i \mid y_i = z_i\}|.$$

Entonces, tenemos que:

$$|D(x,y)| \leq |\{i \mid x_i = z_i\}^c \cup \{i \mid y_i = z_i\}^c| = |\{i \mid x_i \neq z_i\} \cup \{i \mid y_i \neq z_i\}|.$$

Y de aquí se sigue que:

$$|D(x,y)| \leq |\{i \mid x_i \neq z_i\}| + |\{i \mid y_i \neq z_i\}|.$$

Por tanto, $d_H(x,y) \leq d_H(x,z) + d_H(z,y)$. □

Definición 2.3. *Para todo vector $x \in \mathbb{F}_q^n$ su soporte, denotado por $supp(x)$, se define como el conjunto de posiciones de x donde hay elementos distintos de cero. Es decir, $supp(x) = \{i \mid x_i \neq 0\}$. El peso de Hamming de un vector $x \in \mathbb{F}_q^n$ es el número de elementos de su soporte, y se denota por $w_H(x)$.*

$$w_H = |supp(x)| = |\{i \mid x_i \neq 0\}|.$$

Definición 2.4. *La distancia mínima de un código $\mathcal{C}$ se define como:*

$$d_H(\mathcal{C}) = \min_{x,y \in \mathcal{C},\, x \neq y} d_H(x,y).$$

El peso mínimo de un código $\mathcal{C}$ se define como:

$$w_H(\mathcal{C}) = \min_{x \in \mathcal{C},\, x \neq 0} w_H(x).$$

2.2. Códigos lineales

Si el alfabeto $\mathcal{A} = \mathbb{F}_q$ es un cuerpo finito, entonces $\mathbb{F}_q^n$ es un espacio vectorial. Esto nos permite trabajar con $\mathcal{C} \subseteq \mathbb{F}_q^n$ subespacio vectorial. Esta estructura algebraica adicional, dotará a los códigos lineales de algunas propiedades particulares.

Definición 2.5. *Un código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ es un subespacio vectorial de $\mathbb{F}_q^n$. La dimensión de $\mathcal{C}$ es su dimensión como $\mathbb{F}_q$-espacio vectorial. Dado un código lineal $\mathcal{C} \subseteq \mathbb{F}_q^n$ con dimensión $\dim(\mathcal{C}) = k$, lo denotaremos como $[n,k]_q$-código, donde n recibe el nombre de longitud de $\mathcal{C}$ (y la denotaremos $n(\mathcal{C})$) y k la dimensión de $\mathcal{C}$ (y la denotaremos $k(\mathcal{C})$). Si además $d_H(\mathcal{C}) = d$, entonces diremos que $\mathcal{C}$ es un $[n,k,d]_q$-código.*

Nota 1 *Sea $\mathcal{C}$ un $[n,k]_q$-código, entonces el número de palabras de $\mathcal{C}$ es: $|\mathcal{C}| = q^k$.*

Lema 2.6. *La distancia mínima de un código lineal coincide con su peso mínimo.*

Demostración. Primero veamos que $d_H(x,y) = w_H(x-y)$. En efecto, observamos que $d_H(x,y) = |\{i \mid x_i \neq y_i\}| = |\{i \mid x_i - y_i \neq 0\}| = w_H(x-y)$.
Sean $x, y \in \mathcal{C}$ dos palabras distintas de $\mathcal{C}$ que están a una distancia $d_H(\mathcal{C})$, entonces:

$$d_H(\mathcal{C}) = d_H(x,y) = w_H(x-y) \geq w_H(\mathcal{C}),$$

ya que $x - y \in \mathcal{C} - \{0\}$ y $w_H(\mathcal{C})$ es el menor peso de $\mathcal{C}$.
Recíprocamente, sabemos que existe una palabra no nula $c \in \mathcal{C} - \{0\}$ que alcanza el peso mínimo, es decir:

$$w_H(\mathcal{C}) = w_H(c) = d_H(c,0) \geq d_H(\mathcal{C}).$$

El resultado se obtiene de ambas desigualdades. □

2.2.1. Matriz Generatriz de códigos lineales

Definición 2.7. *Sea $\mathcal{C}$ un $[n,k]_q$-código lineal y sea $B = \{g_1, \ldots, g_k\}$ con $g_i = \{g_{i1}, \ldots, g_{in}\} \in \mathbb{F}_q^n$, una base de $\mathcal{C}$ como $\mathbb{F}_q$-espacio vectorial, se denomina matriz generatriz de $\mathcal{C}$ a la matriz G de tamaño $k \times n$ con filas $g_1, \ldots, g_k$, es decir,*

$$G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix} = \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{k1} & \cdots & g_{kn} \end{pmatrix} \in \mathbb{F}_q^{k \times n}.$$

Nota 2 *Dado que B es una base de $\mathcal{C}$, tenemos que G tiene rango k. Por tanto, tras hacer operaciones elementales por filas podemos extraer de G una submatriz identidad I_k de tamaño $k \times k$. Si tras estas operaciones elementales por filas aplicamos permutaciones de columnas entonces una permutación de G tendrá la forma $\big(I_k \mid A\big)$ con $A \in \mathbb{F}_q^{k \times (n-k)}$. Si las columnas $j_1, \ldots, j_k$ de G forman I_k entonces diremos que G está en forma sistemática en las posiciones $(j_1, \ldots, j_k)$.*

Nota 3 *Como las bases de $\mathcal{C}$ no son únicas, las matrices generatrices tampoco lo son.*

La matriz generatriz nos permite definir *el proceso de codificación*. El proceso de codificación, convertir mensajes $m \in \mathbb{F}_q^k$ en palabras del código $c \in \mathcal{C} \subseteq \mathbb{F}_q^n$, de códigos lineales se puede ver como una transformación lineal:

$$\begin{aligned} Enc : \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ m &\longmapsto mG = m_1 g_1 + \cdots + m_k g_k \end{aligned}$$

donde $G = \begin{pmatrix} g_1 \\ \vdots \\ g_k \end{pmatrix} \in \mathbb{F}_q^{k \times n}$.

Si G es sistemática en las posiciones $j_1, \ldots, j_k$, entonces $c_{ij} = m_{ij}$ para todo $i \in \{1, \ldots, k\}$, por tanto el mensaje original aparece en las posiciones $j_1, \ldots, j_k$. En este caso se llama codificación sistemática en las posiciones $(j_1, \ldots, j_k)$.

2.2.2. Matriz de Paridad de códigos lineales

Un espacio vectorial se puede describir de forma paramétrica, es decir, indicando una base del espacio vectorial o implícitamente, es decir, como el conjunto de soluciones de un sistema lineal homogéneo. Como ya vimos anteriormente, la matriz generatriz de un código lo describe en paramétricas. La descripción de las ecuaciones implícitas de un código lineal viene representada por su matriz de paridad.

Definición 2.8. *Sea $H \in \mathbb{F}_q^{(n-k)\times n}$ una matriz de rango $n-k$. Se dice que H es matriz de paridad de un $[n,k]_q$-código si $\mathcal{C}$ es el espacio de soluciones del sistema homogéneo definido por H, es decir:*

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

Observemos que esta matriz nos permite comprobar si una palabra está o no en el código, por eso también se conoce como matriz de control, más concretamente, $x \in \mathcal{C}$ si y solo si $Hx^T = 0$.

El siguiente resultado nos permite relacionar matrices generatrices y de paridad de un código lineal.

Proposición 2.9. *Sean $G \in \mathbb{F}_q^{k\times n}$ una matriz de rango k y $H \in \mathbb{F}_q^{(n-k)\times n}$ una matriz de rango $n-k$, entonces G y H son una matriz generatriz y de paridad de $\mathcal{C}$, un $[n,k]_q$-código lineal, si y solo si $HG^T = 0$.*

Demostración. Supongamos que G es matriz generatriz de un código $\mathcal{C}$ y H matriz de paridad de $\mathcal{C}$, entonces se tiene:

$$\begin{cases} Hc^T = 0 & , \forall c \in \mathcal{C}. \\ mG \in \mathcal{C} & , \forall m \in \mathbb{F}_q^k. \end{cases}$$

Luego,

$$HG^T m^T = H(mG)^T = Hc^T = 0 \ , \ \forall m \in \mathbb{F}_q^k.$$

Por tanto, $HG^T = 0$.

Recíprocamente, como G es una matriz de rango k, G es una matriz generatriz de un código $\mathcal{C}_1$ de parámetros $[n,k]_q$. Como H es una matriz de paridad de rango $n-k$, H es una matriz de paridad de un código $\mathcal{C}_2$ de parámetros $[n,k]_q$. Además para cualquier palabra $c \in \mathcal{C}_1$, $c = mG$ con $m \in \mathbb{F}_q^k$. Luego, $Hc^T = H(mG)^T = HG^T m = 0$. Por lo tanto, $\mathcal{C}_1 \subseteq \mathcal{C}_2$. Como además $\dim(\mathcal{C}_1) = \dim(\mathcal{C}_2)$ se deduce que $\mathcal{C}_1 = \mathcal{C}_2$.

□

Lema 2.10. *Sea $\mathcal{C}$ un $[n,k]_q$-código e I_k la matriz identidad de orden k. Sea $P \in \mathbb{F}_q^{k\times(n-k)}$, entonces $G = \begin{pmatrix} I_k \mid P \end{pmatrix}$ es una matriz generatriz de $\mathcal{C}$ si y solo si $H = \begin{pmatrix} -P^T \mid I_{n-k} \end{pmatrix}$ es una matriz de paridad de $\mathcal{C}$.*

Demostración. $G = \begin{pmatrix} I_k \mid P \end{pmatrix}$ y $H^T = \begin{pmatrix} -P \\ I_{n-k} \end{pmatrix}$. De aquí, vemos que:

$$GH^T = I_k(-P) + PI_{n-k} = -P + P = 0.$$

Aplicando la Proposición 2.9, queda probado el lema. □

En el siguiente resultado veremos que la matriz de paridad nos aporta información sobre la distancia mínima de un código.

Proposición 2.11. *Sea $H \in \mathbb{F}_q^{(n-k)\times n}$ una matriz de paridad de un código $\mathcal{C}$, entonces $d := d_H(\mathcal{C})$, es el menor entero positivo tal que hay d columnas de H linealmente dependientes.*

Demostración. Sea $H = \begin{pmatrix} h_1 \cdots h_n \end{pmatrix} \in \mathbb{F}_q^{(n-k)\times n}$ con i-ésima columna $h_i^T = (h_{i_1}, \ldots, h_{i_{n-k}}) \in \mathbb{F}_q^{n-k}$. Sea $d = d_H(\mathcal{C})$ y $c \in \mathcal{C}$ una palabra no nula de peso mínimo, es decir, $w_H(c) = d$, con $\text{supp(c)} = \{i \mid c_i \neq 0\} = \{j_1, \ldots, j_d\} \subseteq \{1, \ldots, n\}$.

Como $Hc^T = 0$, entonces $c_{j_1}h_{j_1} + \cdots + c_{j_d}h_{j_d} = 0$ y, por tanto, las columnas $\{j_1, \ldots, j_d\}$ son linealmente independientes. Sea $\overline{d}$ es el menor número de columnas linealmente dependientes de H. Hemos visto que $\overline{d} \leq d$.

Recíprocamente, sean $\{h_{j_1}, \ldots, h_{j_{\overline{d}}}\}$ columnas linealmente dependientes con $\overline{d}$ lo más pequeño posible. Entonces, existen $a_1, \ldots, a_{\overline{d}} \in \mathbb{F}_q$ no todos ceros, tales que $a_1h_{j_1} + \cdots + a_{\overline{d}}h_{j_{\overline{d}}} = 0$. Tomamos $c = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ con $c_i = 0$, si $i \notin \{j_1, \ldots, j_{\overline{d}}\}$ y $c_i = a_i$, si $i \in \{j_1, \ldots, j_{\overline{d}}\}$. Entonces de la elección de c se tiene que $Hc^T = 0$, luego $c \in \mathcal{C}$, de donde se deduce que:

$$w_H(\mathcal{C}) = \overline{d} \geq d.$$

□

Corolario 2.12. (Cota de Singleton). *Sea $\mathcal{C}$ un $[n,k,d]_q$-código lineal. Entonces se tiene que: $d \leq n - k + 1$.*

Demostración. Sea $H \in \mathbb{F}_q^{(n-k)\times n}$ una matriz de paridad de $\mathcal{C}$. Por la Proposición 2.11 tenemos que d es el menor entero tal que hay d columnas linealmente dependientes en H. Entonces se tiene que $\text{rang}(H) \geq d-1$. Como el rango de la matriz H es $n-k$ se deduce que $n-k \geq d-1$. □

Definición 2.13. *Un código $\mathcal{C}$ se dice MDS, máxima distancia separable, si su distancia mínima coincide con su Cota de Singleton (Corolario 2.12), es decir,*

$$d(\mathcal{C}) = n(\mathcal{C}) - k(\mathcal{C}) + 1.$$

2.2.3. Decodificación y Capacidad correctora de un código

Definición 2.14. *Sea $\mathcal{C}$ un $[n,k,d]_q$-código lineal. Si $c \in \mathcal{C}$ es la palabra enviada y $r \in \mathbb{F}_q^n$ es el vector recibido, entonces:*

$$E = \{i \mid r_i \neq c_i\},$$

es el conjunto de posiciones de error. Si definimos $e = r - c$, entonces e es el vector error y al valor e_i se le llama valor del error en la posición i.

Definición 2.15. *Un decodificador del código $\mathcal{C}$ por mínima distancia que corrige s errores es una aplicación:*

$$Dec : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k \cup \{?\},$$

tal que si

$$Enc : \mathbb{F}_q^k \longrightarrow \mathcal{C} \subseteq \mathbb{F}_q^n,$$

es un codificador para el código $\mathcal{C}$, entonces $Dec(Enc(m+e)) = m$, para todo $e \in \mathbb{F}_q^n$ con $w_H(e) \leq s$.

Si $Dec(Enc(m)) =?$ entonces se ha producido un fallo en la decodificación. Si $Dec(Enc(m)) = m' \neq m$ entonces se dice que se han producido más de s errores durante la transmisión. Un fallo en la decodificación se puede detectar pero un error en la decodificación no siempre se detecta.

Un decodificador completo es aquel que no tiene fallos, siempre devuelve un mensaje $m' \in \mathbb{F}_q^k$. Un decodificador por mínimas distancias siempre devuelve $m \in \mathbb{F}_q^k$ tal que $Enc(m) = c$ es la palabra más cercana en términos de distancia de Hamming (si hubiese varias palabras a la misma distancia mínima de c, entonces el decodificador devuelve una de ellas) al vector recibido $r \in \mathbb{F}_q^n$.

Definición 2.16. *Diremos que un código $\mathcal{C}$ es l-corrector si para todo $c_1, c_2 \in \mathcal{C}$ dos palabras distintas del código y para todo $e_1, e_2 \in \mathbb{F}_q^n$, con $w_H(e_i) \leq l$ se tiene que $c_1 + e_1 \neq c_2 + e_2$. Es decir, si el número de posiciones de error es a lo sumo l, entonces existe una única palabra del código $c \in \mathcal{C}$ a distancia menor o igual a l de la palabra recibida.*

Definición 2.17. *Se define la capacidad correctora de un código $\mathcal{C}$ como el mayor valor l tal que $\mathcal{C}$ es l-corrector, y se denota por $t_{\mathcal{C}}$.*

Proposición 2.18. *Sea $\mathcal{C}$ un $[n,k,d]_q$-código. Entonces $t_{\mathcal{C}} = \left\lfloor \frac{d-1}{2} \right\rfloor$ es la capacidad correctora de $\mathcal{C}$.*

Demostración. Supongamos que $t_{\mathcal{C}} \leq \lfloor \frac{d-1}{2} \rfloor$. Procedemos por reducción al absurdo suponiendo que $\mathcal{C}$ no es $t_{\mathcal{C}}$-corrector. Es decir, suponemos que existen $c_1, c_2 \in \mathcal{C}$, $e_1, e_2 \in \mathbb{F}_q^n$ tales que $c_1 + e_1 = c_2 + e_2$ con $w_H(e_1), w_H(e_2) \leq t_{\mathcal{C}}$. Entonces, se tiene que, $e_1 - e_2 = c_1 - c_2 \in \mathcal{C}$ y $w_H(c_1 - c_2) = w_H(e_1 - e_2) \leq w_H(e_1) + w_H(e_2) \leq 2t_{\mathcal{C}} < d - 1$, contradiciendo que la distancia mínima de $\mathcal{C}$ sea d.

Recíprocamente supongamos que $\mathcal{C}$ es $t_{\mathcal{C}}$-corrector. Procedemos por reducción al absurdo suponiendo que $d < 2t_{\mathcal{C}}$. Entonces existen $c_1, c_2 \in \mathcal{C}$ que verifican:

$$d = d_H(\mathcal{C}) = d_H(c_1, c_2) = |\{i \mid c_{1i} \neq c_{2i}\}| = |\{i_1, \ldots, i_{N_1}\} \cup \{j_1, \ldots, j_{N_2}\}|,$$

con $N_1, N_2 \leq t_{\mathcal{C}}$.

Definimos $z = (z_1, \ldots, z_n) \in \mathbb{F}_q^n$ tal que $z_i = \begin{cases} z_i = c_{1i} = c_{2i}, & si\ c_{1i} = c_{2i}. \\ z_i = c_{1i}, & si\ i \in \{i_1, \ldots, i_{N_1}\}. \\ z_i = c_{2i}, & si\ i \in \{j_1, \ldots, j_{N_2}\}. \end{cases}$

De esta forma se tiene que:

$$d_H(c_1, z) = N_2 \leq t_{\mathcal{C}} \text{ y } d_H(c_2, z) = N_1 \leq t_{\mathcal{C}}.$$

contradiciendo que $\mathcal{C}$ es $t_{\mathcal{C}}$-corrector.

□

Proposición 2.19. *Todo código $\mathcal{C}$ tienen un decodificador por mínima distancia que corrige $t_{\mathcal{C}}$ errores.*

Demostración. Para ello vamos a describir dicho decodificador. Para cada palabra $r \in \mathbb{F}_q^n$ que se recibe se realiza el siguiente proceso:

1. Se calcula la distancia de Hamming con todas las palabras de $\mathcal{C}$. Es decir $d_H(r, c)$ para todo $c \in \mathcal{C}$.
2. Se elige aquella que minimiza la distancia de Hamming. Es decir:

$$\mathrm{Dec}(r) = \{c \in \mathcal{C} \mid d_H(r, c) = \min_{c \in \mathcal{C}} \{d_H(r, c)\}\}.$$

Si se han cometido menos de $t_{\mathcal{C}}$ errores, entonces el algoritmo devolverá una única palabra, en otro caso devolverá un error ya que habrá más de una palabra que minimice la distancia. □

Sin embargo este algoritmo es muy ineficiente ya que habría que comparar la distancia de Hamming de la palabra recibida con todas las del código. Es decir, sea $\mathcal{C}$ un $[n,k]_q$ código, el algoritmo anterior requiere comparar el vector recibido $r \in \mathbb{F}_q^n$ con las q^k palabras de $\mathcal{C}$, lo que implica realizar $\mathcal{O}(nq^k)$ operaciones. Por esto, uno de los problemas que más ocupan la atención de los investigadores en el área de Teoría de Códigos es la búsqueda de algoritmos eficientes de decodificación de códigos lineales.

En la literatura, todos los algoritmos de decodificación que se han presentado que se pueden utilizar para cualquier código lineal tienen un coste computacional exponencial. Es más, en [2] se demuestra que el problema de decisión de decodificar códigos lineales está catalogado como un problema NP, en este resultado se demuestra que dicho problema es equivalente al problema de emparejamiento 3-dimensional, que se conoce que es un problema NP.

Es por ello, que los investigadores no tienen esperanzas en encontrar algoritmos de decodificación eficientes que funcionen para cualquier código lineal. Sin embargo, sí hay familias de códigos lineales específicas con algoritmos de decodificación rápidos. En el capítulo tres estudiaremos algunas de estas familias y presentaremos en el capítulo cuatro, algoritmos de decodificación eficientes que funcionan sólo en estas familias particulares.

3

Códigos afines

Un código afín o de evaluación es un tipo de código lineal con interesantes propiedades y aplicaciones en diferentes áreas como la criptografía y la compartición de secretos. Los códigos afines fueron introducidos por Fitzgerald y Lax en 1998 [6] y han sido trabajados en detalle por O. Geil y sus coautores [7]. Toda la Teoría de Códigos algebraica está basada en la estructura de cuerpos finitos que vimos en el primer capítulo, todo lo que necesitemos para hablar de códigos afines será visto en este capítulo.

El capítulo comienza con la definición de código afín y el estudio de sus parámetros (longitud, dimensión y distancia mínima, que denotamos $n(\mathcal{C})$, $k(\mathcal{C})$ y $d(\mathcal{C})$, respectivamente). Una buena parte del capítulo (Sección 3.3) está dedicada a definir una cota para la distancia mínima de esta familia de códigos, lo que se conoce como la función huella (footprint bound). Además, definiremos algunas familias notables de códigos afines, como son los códigos Reed-Solomon, los códigos Reed-Muller, los códigos Hiperbólicos y los códigos Cubo. En el final del capítulo nos centraremos en estudiar los parámetros de estas familias de códigos notables.

Cabe destacar que esta cota se presenta en la literatura utilizando herramientas algebraicas como bases de Gröbner, pero en este capítulo hemos hecho un esfuerzo para no necesitar esta herramienta en la construcción de la demostración.

3.1. El conjunto de soluciones de un sistema de ecuaciones

Sea $\mathbb{K}$ un cuerpo cualquiera y $\mathbb{K}[x_1, \ldots, x_m]$ el anillo de polinomios en m variables. Se considera el siguiente sistema de ecuaciones:

$$\begin{cases} h_1(x_1,\ldots,x_m) & = 0 \\ \quad\vdots & \quad\vdots \\ h_s(x_1,\ldots,x_m) & = 0 \end{cases} \tag{3.1}$$

donde $h_1,\ldots,h_s \in \mathbb{K}[x_1,\ldots,x_m]$. Decimos que $(\alpha_1,\ldots,\alpha_m) \in \mathbb{K}^m$ es solución del sistema si y solo si $h_i(\alpha_1,\ldots,\alpha_m) = 0$, para todo $i \in \{1,\ldots,s\}$. Consideramos el ideal $I = (h_1,\ldots,h_s) \subseteq \mathbb{K}[x_1,\ldots,x_m]$.

El objetivo de esta sección es proporcionar una cota para cardinal del conjunto de soluciones del sistema (3.1), que nos será de utilidad a la hora de construir códigos afines. En la prueba de este resultado usaremos el siguiente lema previo.

Lema 3.1. *Sea $\{Q_1,\ldots,Q_r,P\} \subseteq \mathbb{K}^m$ un conjunto de $r+1$ puntos diferentes de $\mathbb{K}^m$. Entonces existe un polinomio $f \in \mathbb{K}[x_1,\ldots,x_m]$ tal que $f(Q_i) = 0$ para todo $i \in \{1,\ldots,r\}$ y $f(P) = 1$.*

Demostración. Sea $Q_i = (\alpha_{i1},\ldots,\alpha_{im}) \in \mathbb{K}^m$, para todo $i \in \{1,\ldots,r\}$ y sea $P = (\beta_1,\ldots,\beta_m) \in \mathbb{K}^m$. Consideramos el polinomio f siguiente:

$$f(x_1,\ldots,x_m) = \prod_{i=1}^{r} \prod_{\substack{1\le i\le r \\ \beta_i \neq \alpha_{ij}}} \frac{x_j - \alpha_{ij}}{\beta_j - \alpha_{ij}}.$$

Se comprueba fácilmente que $f(Q_i) = 0$ para todo $i \in \{1,\ldots,r\}$ y que $f(P) = 1$. □

Proposición 3.2. *Sean $I = (h_1,\ldots,h_s) \subseteq \mathbb{K}[x_1,\ldots,x_m]$ un ideal y $Q_1,\ldots,Q_r \in \mathbb{K}^m$ soluciones del sistema de ecuaciones (3.1). Entonces,*

$$\begin{aligned} \varphi : \mathbb{K}[x_1,\ldots,x_m]/I &\longrightarrow \mathbb{K}^r \\ f + I &\longrightarrow \varphi(f+I) = (f(Q_1),\ldots,f(Q_r)) \end{aligned}$$

es un epimorfismo de espacios vectoriales (aplicación lineal sobreyectiva). En consecuencia, $\dim(\mathbb{K}[x_1,\ldots,x_m]/I) \ge r$.

Demostración. Veamos que φ es aplicación, para ello únicamente falta discutir la unicidad de la imagen. Sean $f,g \in \mathbb{K}[x_1,\ldots,x_m]$ tales que $f + I = g + I$, entonces $f - g \in I$, luego:

$$f - g = \sum_{i=1}^{m} q_i h_i.$$

donde $q_i \in \mathbb{K}[x_1,\ldots,x_m]$. Entonces, como Q_i es una solución del sistema (3.1) para todo $i \in \{1,\ldots,r\}$, se tiene que:

$$(f-g)(Q_i) = \sum_{j=1}^{m} q_j h_j(Q_i) = 0,$$

y por tanto, se tiene que $\varphi(f+I) = \varphi(g+I)$.

Además, se observa que φ es aplicación lineal. Veamos que φ también es sobreyectiva. Para ello basta demostrar que cada uno de los elementos de la base canónica de $\mathbb{K}^r$ pertenecen a $\mathrm{Im}(\varphi)$. Sea e_i el i-ésimo vector de la base canónica, es decir, el vector cuyas entradas son todas nulas salvo en la posición i-ésima, que es 1. En el Lema 3.1 probamos la existencia de un polinomio $f_i \in \mathbb{K}[x_1, \ldots, x_m]$ tal que $f_i(Q_i) = 1$ y $f_i(Q_j) = 0$ para todo $j \in \{1, \ldots, r\} \setminus \{i\}$, por tanto $\varphi(f_i) = e_i$. □

Este resultado aporta una cota sobre el número de soluciones de un sistema de ecuaciones polinomiales. Cuando $\mathbb{K}$ es un cuerpo algebraicamente cerrado, el número de soluciones del sistema (3.1) es exactamente la dimensión de $\mathbb{K}[x_1, \ldots, x_m]/r(I)$ donde $r(I)$ denota al ideal radical de I, es decir,

$$r(I) = \{f \in \mathbb{K}[x_1, \ldots, x_m] \mid \exists n \geq 0 \textit{ tal que } f^n \in I\}.$$

Esta igualdad se puede demostrar con el Teorema de los ceros de Hilbert, cuya demostración está incluida en [5, Theorem 4.1.2]. No obstante, en esta memoria solo necesitamos el resultado de la Proposición 3.2.

3.2. Introducción a código afín

Los códigos afines son códigos lineales y, como tales, se pueden describir como la imagen de un monomorfismo $f : U \hookrightarrow \mathbb{F}_q^n$, donde U es un $\mathbb{F}_q$-espacio vectorial de dimensión igual a la del código en cuestión.

Sea $\mathbb{F}_q$ un cuerpo finito y $m \in \mathbb{Z}^+$, vamos a describir todos los códigos afines sobre $\mathbb{F}_q$ de longitud $n = q^m$. Para ello consideramos $\{P_1, \ldots, P_n\}$ el conjunto de todos los puntos de $\mathbb{F}_q^n$ y el homomorfismo evaluación que evalúa los polinomios de $\mathbb{F}_q[x_1, \ldots, x_m]$ en todos los puntos de $\mathbb{F}_q^n$, es decir,

$$\begin{aligned} \mathrm{ev} : \mathbb{F}_q[x_1, \ldots, x_m] &\longrightarrow \mathbb{F}_q^n \\ f &\longrightarrow \mathrm{ev}(f) = (f(P_1), \ldots, f(P_n)) \end{aligned}$$

El homomorfismo evaluación ev no es inyectivo puesto que $x_i^q - x_i$ es un polinomio que se anula en todos los puntos de $\mathbb{F}_q^n$, para todo $i \in \{1, \ldots, m\}$. En particular, tenemos que el ideal $I = (x_1^q - x_1, \ldots, x_m^q - x_m)$ es un subconjunto de $\mathrm{Ker}(\mathrm{ev})$. Nuestro primer objetivo en esta sección es demostrar que, en efecto, $I = \mathrm{Ker}(\mathrm{ev})$.

Proposición 3.3. *Sean $\mathbb{F}_q$ un cuerpo finito y $m \in \mathbb{Z}^+$. Además, sea I el ideal $I = (x_1^q - x_1, \ldots, x_m^q - x_m) \subseteq \mathbb{F}_q[x_1, \ldots, x_m]$ y $\{P_1, \ldots, P_n\}$ el conjunto de todos los puntos de $\mathbb{F}_q^n$, entonces:*

$$\begin{aligned} \text{ev} : \mathbb{F}_q[x_1, \ldots, x_m]/I &\longrightarrow \mathbb{F}_q^n \\ f + I &\longrightarrow \text{ev}(f + I) = (f(P_1), \ldots, f(P_n)) \end{aligned}$$

es un isomorfismo de espacios vectoriales. En particular, se tiene que:

$$\dim(\mathbb{F}_q[x_1, \ldots, x_m]/I) = n.$$

Demostración. Sabemos que $I \subseteq \text{Ker}(\text{ev})$. Además, por la Proposición 3.2 tenemos que ev es un epimorfismo de espacios vectoriales. Para nuestro resultado solo falta ver que ev es inyectiva y para ello vamos a demostrar que $\dim(\mathbb{F}_q[x_1, \ldots, x_m]/I) \leq n$. Esto lo demostraremos encontrando un sistema generador de $\mathbb{F}_q[x_1, \ldots, x_m]/I$ formado por n elementos. Más concretamente, un sistema generador es $B = \{x_1^{j_1} \cdots x_m^{j_m} + I \mid 0 \leq j_1, \ldots, j_m < q\}$, donde $g + I$ denota la clase de equivalencia módulo I del polinomio $g \in \mathbb{F}_q[x_1, \ldots, x_m]$.

Como todo polinomio $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ es una combinación lineal de monomios de la forma $x_1^{i_1} \cdots x_m^{i_m}$ bastará comprobar que estos últimos son combinación lineal de elementos de B. En efecto, usando la división euclídea, $x_j^{i_j} = q_j(x_j^q - x_j) + r_j$ donde $q_j, r_j \in \mathbb{F}_q[x_j]$ son polinomios en la variable x_j y $\deg(r_j) < q$. De aquí, llevando la expresión al conjunto cociente $\mathbb{F}_q[x_1, \ldots, x_m]/I$, se tiene que: $x_j^{i_j} + I = (q_j(x_j^q - x_j) + I) + (r_j + I)$ y como $(x_j^q - x_j) \in I$ entonces $x_j^{i_j} + I = r_{i_j} + I$.

Luego,

$$\sum_{0 \leq i_1, \ldots, i_m < q} x_1^{i_1} \cdots x_m^{i_m} + I = \sum_{0 \leq i_1, \ldots, i_m < q} r_{i_1} \cdots r_{i_m} + I,$$

es combinación lineal de los elementos de B. Luego, se tiene que:

$$\dim(\mathbb{F}_q[x_1, \ldots, x_m]/I) \leq n.$$

Ahora, utilizando $I \subseteq \text{Ker}(f)$ se deduce que $\dim(\mathbb{F}_q[x_1, \ldots, x_m]/I) = n$. □

En particular de la Proposición 3.3 y su demostración se deduce el siguiente resultado.

Corolario 3.4. *El conjunto* $B = \{x_1^{j_1} \cdots x_m^{j_m} + I \mid 0 \leq j_1, \ldots, j_m < q\}$ *es una base del espacio vectorial* $\mathbb{F}_q[x_1, \ldots, x_m]/I$.

Demostración. En la demostración de la Proposición 3.3 se demuestra que $\{x_1^{j_1} \cdots x_m^{j_m} + I / 0 \leq j_1, \ldots, j_m < q\}$ es un sistema generador de $\mathbb{F}_q[x_1, \ldots, x_m]/I$ y además tiene $n = \dim(\mathbb{F}_q^n)$ elementos, luego B es una base de $\mathbb{F}_q[x_1, \ldots, x_m]/I$. □

Como ev es isomorfismo de espacios vectoriales (Proposición 3.3), al restringir ev a un subespacio vectorial U de $\mathbb{F}_q[x_1, \ldots, x_m]$ se tiene que $\text{ev}_{|U}$ es un monomorfismo de espacios vectoriales. Los códigos afines se definen como las imágenes de los monomorfismos que se obtienen al elegir U como el subespacio generado por un subconjunto L de la base B descrita en el Corolario 3.4. Por tanto, ya podemos definir los códigos afines.

Definición 3.5. *Dado* $L \subseteq B = \{x_1^{j_1} \cdots x_m^{j_m} + I \mid 0 \leq j_1, \ldots, j_m < q\}$ *no vacío, se define el código afín* $\mathcal{C}_L$ *como la imagen de la aplicación:*

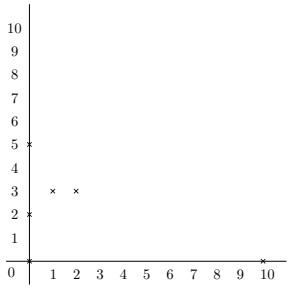
$$\begin{aligned} \text{ev} : \langle L \rangle &\longrightarrow \mathbb{F}_q^n \\ f + I &\longrightarrow (f(P_1), \ldots, f(P_n)) \end{aligned}$$

donde $\mathbb{F}_q^n = \{P_1, \ldots, P_n\}$. *Como ev es aplicación lineal, se tiene que* $\mathcal{C}_L$ *es un* $[n, k, d]_q$*-código con* $n(\mathcal{C}_L) = q^m$ *y* $k(\mathcal{C}_L) = |L|$.

3.2.1. Algunas familias de códigos afines notables

En esta sección vamos a estudiar algunas familias de códigos afines que son muy conocidas en la comunidad de códigos correctores. En particular estudiaremos los códigos Reed-Solomon (que se definen como evaluación de polinomios en una variable), los códigos Reed-Muller (que se definen como evaluación de polinomios en varias variables), los códigos Hiperbólicos (que como veremos más adelante, fueron creados con la intención de ser los códigos con mayor dimensión fijada una distancia mínima) y los códigos Cubo (que se pueden ver como el producto de códigos Reed-Solomon).

Nota 4 *Si identificamos el monomio* $x_1^{\alpha_1} \cdots x_m^{\alpha_m} \in L$ *con el punto de coordenadas* $(\alpha_1, \ldots, \alpha_m) \in \mathbb{N}^m$*, entonces los códigos afines* $\mathcal{C}_L$ *están en biyección con los subconjuntos no vacíos de* $\{1, \ldots, q-1\}^m$*, (ver Figura 3.1a para un ejemplo con* $m = 2$*). Haremos uso de esta biyección y, abusando de notación, escribiremos* $\mathcal{C}_L$ *siendo* $L \subseteq \{1, \ldots, q-1\}^m$.



(a) Representación gráfica del conjunto:

$$L = \{(0,0), (2,0), (3,1), (3,2), (5,0), (0,10)\},$$

cada × representa un elemento del conjunto L.

Figura 3.1

Veamos ahora algunas familias de códigos afines notables.

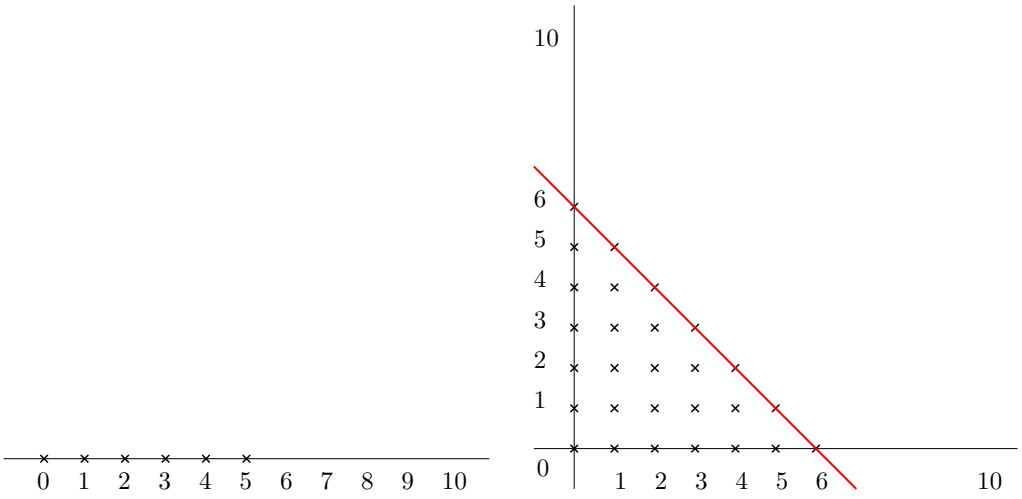
Los códigos Reed-Solomon fueron presentados por S.Reed y G.Solomon en 1960 [18]. Estos códigos correctores son muy utilizados en dispositivos como CD's, DVD's, Blu-Ray, DSL, WIMAX o RAID.

Definición 3.6. *Sea* $t \in \mathbb{Z}^+$ *tal que* $0 \leq t \leq q-1$ *y* $L_{RS} = \{\alpha \mid 0 \leq \alpha \leq t\}$. *Entonces a* $\mathcal{C}_{L_{RS}}$ *se le llama código Reed-Solomon de grado* t *y se denota por* $RS_q(t)$.

Los códigos Reed-Muller fueron introducidos como una generalización de los códigos Reed-Solomon en 1954 por D.E.Muller [14] y S.Reed [17] quien propuso el primer algoritmo de decodificación eficiente en el caso binario.

Definición 3.7. *Sean* $t, m \in \mathbb{Z}^+$ *con* $0 \leq t \leq m(q-1)$ *y además sea el conjunto* $L_{RM} = \{(\alpha_1, \ldots, \alpha_m) \mid 0 \leq \alpha_1, \ldots, \alpha_m \leq t, \sum_{i=1}^{m} \alpha_i \leq t\}$. *Entonces a* $\mathcal{C}_{L_{RM}}$ *se le llama código Reed-Muller de grado* t *en* m *variables y se denota por* $RM_q(t, m)$.

Se observa que los códigos Reed-Solomon son exactamente códigos Reed-Muller con $m = 1$.



(a) Representación gráfica del conjunto L_{RS} que define al código Reed-Solomon, $RS_{11}(5)$.

(b) Representación gráfica del conjunto L_{RM} que define al código Reed-Muller, $RM_{11}(6, 2)$.

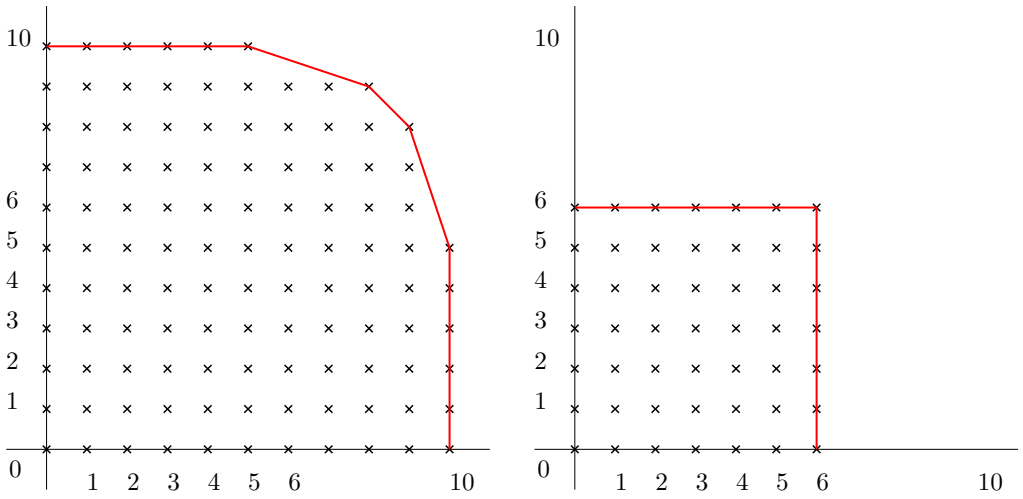
Figura 3.2

Los códigos Hiperbólicos fueron definidos por primera vez por Saints y Heegard en 1993 como una mejora de los códigos Reed-Muller $RM_q(t, m)$, aunque en este momento no se les conocía por ese nombre. Esta construcción fue generalizada para un m arbitrario por Feng y Rao que estimaron la distancia mínima de de estos códigos. Más tarde, se acotó su distancia mínima utilizando técnicas algebraicas potentes como las bases de Gröbner y fue en ese momento cuando se renombraron como códigos Hiperbólicos (véase [8] para más detalles).

Definición 3.8. *Sean* $\delta, m \in \mathbb{Z}^+$ *con* $1 \leq \delta \leq q^m$ *y además sea el conjunto* $L_H = \{(\alpha_1, \ldots, \alpha_m) \mid 0 \leq \alpha_1, \ldots, \alpha_m < q, (q-\alpha_1)\cdots(q-\alpha_m) \geq \delta\}$. *Entonces a* $\mathcal{C}_{L_H}$ *se le llama código hiperbólico de grado* δ *y se denota por* $Hyp_q(\delta, m)$.

Los códigos Cubo fueron introducidos por Parvaresh, El-Khamy, Stepanov, Augot, McEliece y Vardy [15] en 2006 como el producto de códigos Reed-Solomon, que puede ser visto como un código afín basado en polinomios con grados acotados. Posteriormente, en el 2018, Kopparty, Ron-Zewi, Saraf y Wootters [13] mostraron que los códigos Cubo optimizaban algunas propiedades relacionadas con la corrección de errores.

Definición 3.9. *Sean* $t, m \in \mathbb{Z}^+$ *con* $0 \leq t < q$ *y además sea el conjunto* $L_C = \{(\alpha_1, \ldots, \alpha_m) \mid 0 \leq \alpha_1, \ldots, \alpha_m \leq t\}$. *Entonces a* $\mathcal{C}_{L_C}$ *se le llama código cubo de grado* t *y se denota por* $Cubo_q(t, m)$.



(a) Representación gráfica del conjunto L_H que define al código Hiperbólico, $Hyp_{11}(6, 2)$.

(b) Representación gráfica del conjunto L_C que define al código Cubo, $Cubo_{11}(6, 2)$.

Figura 3.3

3.3. Función huella: Una cota para la distancia de códigos afines

El objetivo de esta sección será presentar una cota inferior para la distancia mínima de un código afín. Los resultados de esta sección se han obtenido de [19], no obstante, presentaremos una prueba alternativa que no hace uso de las bases de Gröbner.

Definición 3.10. *Sea $L \subseteq \{0,\dots,q-1\}^m$ definimos como función huella del código afín $\mathcal{C}_L$ al entero:*

$$\mathrm{FB}(\mathcal{C}_L) = \min_{\beta_1,\dots,\beta_m \in L}\{(q-\beta_1)\cdots(q-\beta_m)\}.$$

Lema 3.11. *Sea*

$$\begin{aligned}\delta : \{0,\dots,q-1\}^m &\longrightarrow \mathbb{N}\\ (\alpha_1,\dots,\alpha_m) &\longrightarrow \delta(\alpha_1,\dots,\alpha_m) = \sum_{i=1}^{m}\alpha_i q^{i-1},\end{aligned}$$

entonces δ es una aplicación inyectiva.

Demostración. δ es inyectiva ya que si $\delta(\alpha_1,\dots,\alpha_m) = \delta(\alpha'_1,\dots,\alpha'_m)$, entonces se tiene que:

$$\sum_{i=1}^{m}\alpha_i\cdot q^{i-1} = \sum_{i=1}^{m}\alpha'_i\cdot q^{i-1}.$$

Ambos sumatorios dan por resultado la expresión de números naturales en base q de la forma $\beta_1+\beta_2 q+\cdots+\beta_m q^{m-1}$ entonces se tendría que $\alpha_i = \alpha'_i\ \forall i = 1,\dots,m$ ya que la expresión de un número en base q es única, por el *Teorema fundamental de la representación en base q* [12, Páginas 194-213]. □

Definición 3.12. *Sea $f = \sum_{0\le j_1,\dots,j_m<q} a_{j_1,\dots,j_m}x_1^{j_1}\cdots x_m^{j_m} \in \mathbb{F}_q[x_1,\dots,x_m]$ un polinomio, se define el soporte de f y se denota por* $\mathrm{supp}(f)$ *al conjunto de los monomios de f cuyo coeficiente es no nulo. Es decir,*

$$\mathrm{supp}(f) = \{x_1^{j_1}\cdots x_m^{j_m} \mid a_{j_1,\dots,j_m} \neq 0\}.$$

Antes de acotar la distancia mínima de un código afín, veamos un lema técnico que nos será de utilidad.

Lema 3.13. *Sea $\mathsf{L}_f = \mathbb{F}_q[x_1,\dots,x_m]/(f, x_1^q-x_1,\dots,x_m^q-x_m)$ un $\mathbb{F}_q$-espacio vectorial, donde $f \in \mathbb{F}_q[x_1,\dots,x_m]$ es no nulo e I es el ideal $(f, x_1^q-x_1,\dots,x_m^q-x_m)$. Tomamos $x_1^{\alpha_1}\cdots x_m^{\alpha_m} \in \mathrm{supp}(f)$ tal que:*

$$\delta(\alpha_1,\dots,\alpha_m) = \max\{\delta(\beta_1,\dots,\beta_m) \mid x_1^{\beta_1}\cdots x_m^{\beta_m} \in \mathrm{supp}(f)\}.$$

Entonces:

$$S = \{x_1^{j_1}\cdots x_m^{j_m} + I \mid 0 \le j_1,\dots,j_m < q \text{ y } x_1^{\alpha_1}\cdots x_m^{\alpha_m} \text{ no divide a } x_1^{j_1}\cdots x_m^{j_m}\},$$

es un sistema generador de L_f, en particular, $\dim(\mathsf{L}_f) \le n-(q-\alpha_1)\cdots(q-\alpha_m)$.

Demostración. Por el Corolario 3.4, $B = \{x_1^{j_1} \cdots x_m^{j_m} + I \mid 0 \leq j_1, \ldots, j_m < q\}$ es un sistema generador de L_f. Veamos que los elementos de B son combinación lineal de elementos de S. Definimos:

$$\begin{aligned} \Delta : B &\longrightarrow \mathbb{N} \\ x_1^{j_1} \cdots x_m^{j_m} + I &\longrightarrow \begin{cases} 0, & si\ x_1^{j_1} \cdots x_m^{j_m} + I \in S. \\ \delta(j_1, \ldots, j_m), & en\ caso\ contrario. \end{cases} \end{aligned}$$

Procedemos ahora por inducción sobre $\Delta(x_1^{j_1} \cdots x_m^{j_m} + I) \in \mathbb{N}$.

Si $\Delta(x_1^{j_1} \cdots x_m^{j_m} + I) = 0$, entonces $x_1^{j_1} \cdots x_m^{j_m} + I \in S$. Supongamos ahora que $\Delta(x_1^{j_1} \cdots x_m^{j_m} + I) > 0$, entonces se tiene que hay puntos en el conjunto $\{\delta(j_1, \ldots, j_m) \mid x_1^{j_1} \cdots x_m^{j_m} \notin S\}$, luego existe un monomio $x_1^{i_1} \cdots x_m^{i_m} \notin S$ tal que $\delta(i_1, \ldots, i_m) = \Delta(x_1^{j_1} \cdots x_m^{j_m} + I)$. Por otro lado, definimos $g = x_1^{i_1} \cdots x_m^{i_m} \notin S$, entonces $x_1^{\alpha_1} \cdots x_m^{\alpha_m}$ divide a $x_1^{i_1} \cdots x_m^{i_m}$.

Consideramos ahora $h = g - \frac{1}{b_{\alpha_1 \cdots \alpha_m}} x_1^{i_1 - \alpha_1} \cdots x_m^{i_m - \alpha_m} f$, es fácil ver que $\overline{h} = \overline{g}$, nuestro objetivo es ver que $\Delta(h_i) < \Delta(g)$, donde $h_i \in \operatorname{supp}(h)$.

Para ver esta desigualdad no hará falta tener en cuenta a g, ya que lo hemos anulado definiendo h. Entonces para probar la desigualdad faltaría ver los monomios de $\frac{1}{b_{\alpha_1 \cdots \alpha_m}} x_1^{i_1 - \alpha_1} \cdots x_m^{i_m - \alpha_m} f$, pero ninguno de estos monomios puede hacer que $\Delta(h_i) = \Delta(g)$, ya que tienen menor grado.

Sin embargo, podríamos pensar que cabe la posibilidad de que haya algún exponente mayor que q, por lo que no se podría aplicar la función δ que está definida para el conjunto $\{0, \ldots, q-1\}^m$. No obstante, estamos trabajando en $\mathbb{F}_q$ y por el Teorema 1.3(6), sabemos que $x_i^q = x_i$, para todo $i \in \{1, \ldots, m\}$. Por tanto, aplicando esta propiedad, de tener un exponente mayor que q podríamos cambiar previamente el monomio y así lograr que su exponente se encuentre en el conjunto $\{0, \ldots, q-1\}^m$ permitiéndonos de esta forma aplicar la función δ.

Luego $\Delta(h_i) < \Delta(g)$, concluyendo así que el valor de Δ disminuye y por consiguiente S es sistema generador de L_f. Por tanto, $\dim(\mathsf{L}_f) \leq |S|$ y además se tiene que $|S| = n - |B \backslash S|$, donde el conjunto $B \backslash S$ representa a los monomios $x_1^{i_1} \cdots x_m^{i_m} \in B$ tales que $x_1^{\alpha_1} \cdots x_m^{\alpha_m}$ divide a $x_1^{i_1} \cdots x_m^{i_m}$. Entonces $i_j \geq \alpha_j$ para todo $j \in \{1, \ldots, m\}$, luego se tiene que $i_j \in \{\alpha_j, \alpha_{j+1}, \ldots, q-1\}$ para todo $j \in \{1, \ldots, m\}$. Concluyendo así que hay $(q - \alpha_j)$ posibles valores para i_j, luego $|B \backslash S| = (q - \alpha_1) \cdots (q - \alpha_m)$.

□

Teorema 3.14. *Sea $\mathcal{C}_L$ un código afín, entonces se tiene que $d(\mathcal{C}_L) \geq \mathrm{FB}(\mathcal{C}_L)$.*

Demostración. Sabemos que la distancia mínima de un código se define como:

$$d(\mathcal{C}_L) = \min_{\substack{\gamma \in \mathcal{C}_L \\ \gamma \neq 0}} \{w_H(\gamma)\} = \min_{\substack{f \in \langle L \rangle \\ f \neq 0}} \{w_H(f(P_1), \ldots, f(P_n)\},$$

donde $\mathbb{F}_q^m = \{P_1, \ldots, P_n\}$. Observamos que $w_H(f(P_1), \ldots, f(P_n))$ es exactamente el número de puntos de $\mathbb{F}_q^m$ que no anulan a f. Por tanto, tenemos que:

$$\min_{\substack{f \in \langle L \rangle \\ f \neq 0}} \{w_H(f(P_1), \ldots, f(P_n))\} = n - \max_{\substack{f \in \langle L \rangle \\ f \neq 0}} |\{\text{Raíces de } f \text{ en } \mathbb{F}_q^m\}|.$$

Además las raíces de f en $\mathbb{F}_q^m$ son exactamente las soluciones del sistema polinomial de ecuaciones siguiente:

$$\begin{cases} f = 0 \\ x_1^q - x_1 = 0 \\ \quad \vdots \\ x_m^q - x_m = 0 \end{cases} \tag{3.2}$$

Obsérvese que f puede tener infinitas raíces $(\alpha_1, \ldots, \alpha_n)$, donde α_j pertenece a la clausura algebraica de $\mathbb{F}_q$ para todo $j \in \{1, \ldots, n\}$, pero al añadir al sistema las ecuaciones $x_i^q - x_i$ para todo $i \in \{1, \ldots, m\}$, garantizamos que cualquier solución del sistema es un elemento de $\mathbb{F}_q^m$, véase el Teorema 1.3(6).

Luego, por la Proposición 3.2 tenemos que:

$$d(\mathcal{C}_L) = n - \max_{\substack{f \in \langle L \rangle \\ f \neq 0}} |\{\text{Raíces de } f \text{ en } \mathbb{F}_q^m\}| =$$

$$= n - \max_{f \in \langle L \rangle} |\{\text{Soluciones del sistema (3.2)}\}| \geq$$

$$\geq n - \max_{f \in \langle L \rangle} \{\dim(\mathbb{F}_q[x_1, \ldots, x_m]/(f, x_1^q - x_1, \ldots, x_m^q - x_m))\},$$

donde f es un polinomio no nulo. Denotamos por L_f al $\mathbb{F}_q$-espacio vectorial $\mathsf{L}_f = \mathbb{F}_q[x_1, \ldots, x_m]/(f, x_1^q - x_1, \ldots, x_m^q - x_m)$. En virtud del Lema 3.13 sabemos que $\dim(\mathsf{L}_f) \leq n - (q - \beta_1) \cdots (q - \beta_m)$. Por tanto:

$$n - \max_{f \in \langle L \rangle} \{\dim(\mathsf{L}_f)\} \geq n - \max_{\beta_1, \ldots, \beta_m \in L} \{n - (q - \beta_1) \cdots (q - \beta_m)\} =$$

$$= \min_{\beta_1, \ldots, \beta_m \in L} \{(q - \beta_1) \cdots (q - \beta_m)\}.$$

Concluyendo así que:

$$d(\mathcal{C}_L) \geq \min_{\beta_1, \ldots, \beta_m \in L} \{(q - \beta_1) \cdots (q - \beta_m)\}.$$

□

3.4. Parámetros de algunas familias de códigos afines notables

Veamos una proposición que nos proporcionará una condición que será de utilidad posteriormente.

Proposición 3.15. *Supongamos que* $\mathrm{FB}(\mathcal{C}_L) = (q-\alpha_1)\cdots(q-\alpha_m)$. *Si para todo* $\beta = (\beta_1, \ldots, \beta_m) \in \mathbb{N}^m$ *con* $0 \leq \beta_i \leq \alpha_i$ *se tiene que* $\beta \in L$, *entonces* $d(\mathcal{C}_L) = \mathrm{FB}(\mathcal{C}_L)$.

Demostración. Sea d la distancia mínima del código. Sabemos por el Teorema 3.14 que $d \geq \mathrm{FB}(\mathcal{C}_L)$. Veamos que, en las condiciones de la proposición, $d \leq \mathrm{FB}(\mathcal{C}_L)$.
Sean $\{P_1, \ldots, P_n\}$ los puntos de $\mathbb{F}_q^m$, definimos:

$$h = \prod_{i=1}^{m} (x_i - P_1)\cdots(x_i - P_{\alpha_i}).$$

El polinomio está bien definido, ya que por hipótesis se tiene que para todo $\alpha = (\overline{\alpha_1}, \ldots, \overline{\alpha_m})$, $0 \leq \overline{\alpha_i} \leq \alpha_i$, se tiene que $\overline{\alpha_i} \in L$, para todo $1 \leq i \leq m$. Además sabemos que $w_H(\mathrm{ev}(h)) = q^m - (\text{número de raíces de } h \text{ en } \mathbb{F}_q^m)$, luego habrá que ver cuántas raíces distintas tiene h. Las raíces de h son de la forma:

$$(P_{i_1}, A_{2_1}, \ldots, A_{m_1}) \ , \ (A_{1_2}, P_{i_2}, \ldots, A_{m_2}) \ , \ \ldots \ , \ (A_{1_m}, \ldots, A_{(m-1)_m}, P_{i_m}),$$

con $1 \leq i_1 \leq \alpha_1$, $\ldots$, $1 \leq i_m \leq \alpha_m$ y $A_{i_j} \in \mathbb{F}_q$ con $1 \leq j \leq m$, $1 \leq i \leq m$, $i \neq j$. Luego, habrán $(q-\alpha_1)\cdots(q-\alpha_m)$ raíces, por tanto, hemos encontrado una palabra del código cuyo peso coincide con la función huella, concluyendo así que $d(\mathcal{C}_L) = \mathrm{FB}(\mathcal{C}_L)$. □

Definición 3.16. *Sea* $\mathcal{M} \subseteq \mathbb{F}_q[x_1, \ldots, x_m]$. *Se dice que el conjunto* $\mathcal{M}$ *es cerrado bajo divisibilidad si de tener* $f \in \mathcal{M}$ *y* $g \in \mathbb{F}_q[x_1, \ldots, x_m]$ *tal que* g *divide a* f, *entonces* $g \in \mathcal{M}$.

La propiedad ser cerrado bajo divisibilidad es más general que la propiedad que aporta la Proposición 3.15, su definición ha sido extraída de [3]. En particular las familias de códigos afines $\mathcal{C}_L$ que hemos visto en la sección 3.2.1, verifican que L es cerrado bajo divisibilidad.

3.4.1. Parámetros de los códigos Reed-Solomon

En esta subsección estudiaremos los parámetros de los códigos Reed-Solomon (ver Definición 3.6).

Teorema 3.17. *Sea $t \in \{0, \ldots, q-1\}$, entonces el código Reed-Solomon $RS_q(t)$ tiene como parámetros $n(RS_q(t)) = q$, $k(RS_q(t)) = t+1$ y $d(RS_q(t)) = q-t$.*

Demostración. Dado que $RS_q(t) = \mathcal{C}_L$ siendo $L = \{0, \ldots, t\}$, se tiene por la Definición 3.6 que la longitud del código es q y la dimensión $t+1$. Además, del Teorema 3.14 se tiene que $d(RS_q(t)) \geq \mathrm{FB}(RS_q(t)) = q-t$ y de la cota de Singleton, Corolario 2.12, se sigue que $d(RS_q(t)) \leq q-t$. Por tanto $d = q-t$. □

3.4.2. Parámetros de los códigos Reed-Muller

En esta subsección estudiaremos los parámetros de los códigos Reed-Muller (ver definición 3.7).

Proposición 3.18. *Dados $t, m \in \mathbb{Z}^+$ tales que $t \leq m(q-1)$. Sean $a, b \in \mathbb{N}$ tales que $t = a \cdot (q-1) + b$ con $0 \leq b \leq q-1$. Entonces el código Reed-Muller $RM_q(t, m)$ tiene como parámetros $n(RM_q(t, m)) = q^m$ y $d(RM_q(t, m)) = (q-b)q^{m-1-a}$.*

Demostración. Sea $L = \{(\alpha_1, \ldots, \alpha_m) \mid 0 \leq \alpha_1, \ldots, \alpha_m \leq t, \sum_{i=1}^m \alpha_i \leq t\}$, entonces, dado que $RM_q(t, m) = \mathcal{C}_L$, tenemos que la longitud del código es q^m. Los códigos Reed-Muller, $RM_q(t, m)$, son cerrados bajo divisibilidad, por la Proposición 3.15 la distancia mínima de esta familia de códigos coincide con su *Footprint Bound*. Calculemos ahora la distancia mínima del código $RM_q(t, m)$.

$$d(RM_q(t, m)) = \mathrm{FB}(RM_q(t, m)) = \min_{\beta_1, \ldots, \beta_m \in L} \{(q-\beta_1) \cdots (q-\beta_m)\}.$$

Por lo tanto calcular la distancia mínima del código es equivalente a minimizar el volumen de un cubo de m-dimensiones con medidas $(q-\beta_1) \times \cdots \times (q-\beta_m)$ con $1 \leq \beta_1, \ldots, \beta_m \leq q-1$. Además, este volumen será mínimo si $\beta_1 + \cdots + \beta_m = t$ ya que, en otro caso, existiría al menos un $i \in \{1, \ldots, m\}$ para el cual existe un β_i' tal que $\beta_i' < \beta_i$, luego se tendría que $q - \beta_i' > q - \beta_i$, aumentando el volumen. Además si $\beta_1 + \cdots + \beta_m = t$, a lo sumo existirá un $l \in \{1, \ldots, q-1\}$ distinto de 1 y $q-1$, ya que en caso contrario existen $i, j \in \{1, \ldots, m\}$ tal que $\beta_i, \beta_j \notin \{1, q-1\}$ y $1 < \beta_i < \beta_j < q-1$, entonces podemos definir $\beta_i' = \beta_i - 1$, $\beta_j' = \beta_j + 1$, con $\beta_i', \beta_j' \in \{1, \ldots, q-1\}$ y $\beta_l' = \beta_l$ para todo $l \in \{1, \ldots, m\} \backslash \{i, j\}$, entonces:

$$\prod_{s=1}^{m} (q - \beta_s') < \prod_{s=1}^{m} (q - \beta_s),$$

pues $(q-\beta_i')(q-\beta_j') < (q-\beta_i)(q-\beta_j)$ ya que $\beta_i - 1 < \beta_j$, por lo tanto disminuiría el volumen.

Entonces, si $t = a(q-1) + b$, $0 \leq b \leq q-1$, el mínimo se obtiene en $(\beta_1, \ldots, \beta_m) \in L$ con $\beta_1 = \cdots = \beta_a = q-1$, $\beta_{a+1} = b$ y $\beta_{a+2} = \cdots = \beta_m = 0$. De aquí:

$$d(\mathcal{C}) = (q-\beta_1)\cdots(q-\beta_a)\cdot(q-\beta_{a+1})(q-\beta_{a+2})\cdots(q-\beta_m) = (q-b)q^{m-1-a}.$$

□

Antes de hablar de la dimensión de los códigos Reed-Muller, veamos un lema técnico.

Lema 3.19. *Sea $I = \{1,\ldots,m\}$, definimos el conjunto:*

$$L_I = \{(\alpha_1,\ldots,\alpha_m) \in \mathbb{N}^m \mid \sum_{i=1}^{m} \alpha_i \leq t \text{ y } \alpha_i \geq q \text{ si } i \in I\}.$$

Entonces $|L_I| = \binom{m+t-|I|q}{m}$ *si* $t \geq |I|q$ *y* $|L_I| = 0$ *en caso contrario.*

Demostración. Por definición, $L_I = \{(\alpha_1,\ldots,\alpha_m) \in \mathbb{N}^m \mid \sum_{i=1}^m \alpha_i \leq t \text{ y } \alpha_1 \geq q\}$. Haciendo el cambio de variable $\alpha_i' = \alpha_i - q$ si $i \in I$ y $\alpha_i' = \alpha_i$ si $i \notin I$, entonces:

$$L_I = \{(\alpha_1',\ldots,\alpha_m') \in \mathbb{N}^m \mid \sum_{i=1}^{m} \alpha_i' \leq t - |I|q\} =$$

$$= \{(\alpha_1',\ldots,\alpha_m',\beta) \in \mathbb{N}^{m+1} \mid \sum_{i=1}^{m} \alpha_i' + \beta = t - |I|q\}$$

luego $|L_I| = \binom{m+t-|I|q}{m}$.

Si $t \geq |I|q$ no hay problemas en el cálculo del coeficiente binomial anterior, pero si $t < |I|q$, entonces $m+t-|I|q < m$ y el coeficiente binomial en este caso es cero, luego $|L_I| = 0$. □

En [11, Proposition 2] se da una demostración de la distancia mínima de los códigos Reed-Muller usando bases de Gröbner.

Proposición 3.20. *Sean $t, m \in \mathbb{Z}^+$ tales que $t \leq m(q-1)$. Entonces la dimensión del código Reed Muller $RM_q(t,m)$ es:*

$$k(RM_q(t,m)) = \sum_{i=0}^{m} (-1)^i \binom{m}{i} \binom{m+t-iq}{m}.$$

Demostración. Podemos relacionar el cardinal del conjunto L_{RM} (ver Definición 3.7) con el cardinal de los conjuntos del Lema 3.19. Usando la notación de este lema se observa que:

$$RM = L_\emptyset - (\cup_{i=1}^m L_i). \tag{3.3}$$

Además, por definición, se tiene que $L_I \cap L_J = L_{I\cup J}$. Ahora, por el Principio de Inclusión-Exclusión, tenemos que:

$$|L_1 \cup \ldots \cup L_m| = \sum_{|I|=1} |L_I| - \sum_{|I_1\cup I_2|=2} L_{I_1\cup I_2} + \cdots + \sum_{|I_1\cup\ldots\cup I_m|=m} (-1)^{m-1} |L_{I_1\cup\ldots\cup I_m}| =$$

$$= \sum_{i=1}^{m} (-1)^{i-1} \binom{m}{i} |L_i|,$$

ahora por el Lema 3.19 y la Ecuación (3.3) se tiene que:

$$|L_{RM}| = \sum_{i=0}^{m} (-1)^i \binom{m}{i} \binom{m+t-iq}{m}.$$

□

3.4.3. Parámetros de los códigos Hiperbólicos

En esta subsección estudiaremos los parámetros de los códigos hiperbólicos (ver Definición 3.8), además veremos que $Hyp_q(t,m)$ es el código afín con mayor dimensión de entre todos los códigos afines $\mathcal{C}_L$ de longitud q^m tales que $\mathrm{FB}(\mathcal{C}_L) \geq t$. Esta familia de códigos es cerrada bajo divisibilidad y por tanto, su distancia mínima coincide son su función huella.

Proposición 3.21. *Sea $\mathcal{C}_L$ un código afín de longitud q^m y sea $t \leq \mathrm{FB}(\mathcal{C}_L)$. Entonces, $k(\mathcal{C}_L) \leq k(Hyp_q(t,m))$.*

Demostración. Si demostramos que $L \subseteq L_H$, se tendría que $k(\mathcal{C}_L) \leq k(Hyp_q(t,m))$. Sea $\gamma \in \mathcal{C}_L$, entonces existe un polinomio $f = \sum_{\alpha_i \in L} x_1^{\alpha_1} \cdots x_m^{\alpha_m}$ tal que $\gamma = \mathrm{ev}(f)$. Pero como $\mathrm{FB}(\mathcal{C}_L) \geq \mathrm{FB}(Hyp_q(t,m))$, entonces si $\alpha_1, \ldots, \alpha_m \in L$, se tiene que $(x_1 - \alpha_1) \cdots (x_m - \alpha_m) \geq \mathrm{FB}(\mathcal{C}_L) \geq \mathrm{FB}(Hyp_q(t,m))$. Luego $\alpha_i \in L_H$ para todo $i \in \{1, \ldots, m\}$, por tanto $L \subseteq L_H$. □

3.4.4. Parámetros de los códigos Cubo

En esta subsección estudiaremos los parámetros de los códigos Cubo (ver Definición 3.9).

Proposición 3.22. *Dados $t \in \{0, \ldots, q-1\}, m \in \mathbb{Z}^+$, entonces el código cubo $Cubo_q(t,m)$ tiene como parámetros $n(Cubo_q(t,m)) = q^m$, $k(Cubo_q(t,m)) = (t+1)^m$ y $d(Cubo_q(t,m)) = (q-t)^m$.*

Demostración. Sea $L = \{(\alpha_1, \ldots, \alpha_m) \mid 0 \leq \alpha_1, \ldots, \alpha_m \leq t\}$, entonces dado que $Cubo_q(t,m) = \mathcal{C}_L$, tenemos que la longitud del código es q^m y la dimensión $(t+1)^m$, por la Definición 3.5. Los códigos Cubo $Cubo_q(t,m)$ son cerrados bajo

divisibilidad, véase Proposición 3.15, por tanto la distancia mínima de esta familia de códigos coincide con su función huella. Calculemos ahora la distancia mínima del código $Cubo_q(t, m)$.

$$d(\mathcal{C}_L) = \min_{\beta_1,\ldots,\beta_m \in L} (q - \beta_1) \cdots (q - \beta_m).$$

Calcular la distancia mínima del código es equivalente a minimizar el volumen de un cubo de m-dimensiones con medidas $(q-\beta_1) \times \cdots \times (q-\beta_m)$. Sabemos que para todo $i = 1, \ldots, m$ se tiene que $\beta_i \in L$, luego $\beta_i \leq t$. Por lo tanto el volumen será mínimo si $\beta_i = t$ para todo $i = 1, \ldots, m$, ya que en caso contrario existiría j con $1 \leq j \leq m$ tal que $\beta_j < t$ luego $(q - t) < (q - \beta_j)$ aumentando el valor del volumen. Luego $d(\mathcal{C}_L) = \mathrm{FB}(\mathcal{C}_L) = (q - t)^m$. □

4

Decodificación eficiente de algunas familias de códigos afines

Como ya comentamos en la Proposición 2.19, se conoce la existencia de un algoritmo decodificador desde un punto de vista teórico, conocido por fuerza bruta el cual consiste en comparar la distancia de Hamming de la palabra recibida con todas las del código, sin embargo esto es muy ineficiente, es por eso que en la práctica el problema radica en encontrar un algoritmo que compute de manera eficiente la salida del algoritmo. En [2] se demuestra que el siguiente problema de decisión es NP-completo:

"Dado un código lineal $\mathcal{C}$ definido por su matriz generatriz $k \times n$ con coeficientes en $\mathbb{F}_q$, un vector $x \in \mathbb{F}_q^n$ y un valor $e \geq 0$, ¿existe una palabra $w \in \mathcal{C}$ tal que $d_H(x, w) \leq e$?".

En vista de este resultado negativo, no es esperable encontrar un algoritmo eficiente (polinomial) que resuelva el problema de decodificación para cualquier código lineal. En consecuencia, los esfuerzos se concentran en obtener algoritmos de decodificación eficientes que funcionen para determinadas familias de códigos lineales.

En concreto, en este capítulo vamos a estudiar algoritmos de decodificación eficientes de algunas familias de códigos afines notables: códigos Reed-Solomon, códigos Cubo y códigos Reed-Muller binarios. En particular vamos a presentar un algoritmo de decodificación del código Reed-Solomon ($RS_q(s)$), que alcanza su capacidad de corrección (Sección 4.1). También presentaremos un algoritmo para los códigos Reed-Muller binarios ($RM_2(r, m)$) que también alcanza su capacidad de corrección (Sección 4.3) y finalmente presentaremos un algoritmo para códigos Cubo ($Cubo_q(s, m)$) pero en este caso no alcanza la capacidad de correctora del código, el número de errores que podemos corregir con este algoritmo es menor que $\frac{d(\mathcal{C})-1}{2}$ (véase Proposición 2.18).

Encontrar familias de códigos con algoritmos de decodificación eficientes es una temática en la que sigue siendo activa la investigación debido a sus múltiples aplicaciones prácticas, por ejemplo en criptografía.

4.1. Decodificación de códigos Reed-Solomon

El objetivo de esta sección será el de proporcionar un algoritmo para la decodificación de códigos Reed-Solomon. En esta memoria adaptaremos el algoritmo Berlekamp – Welch que presentaron Berlekamp y Welch en [1] en el año 1968.

Sea α un elemento primitivo de $\mathbb{F}_q$, es decir, un generador del grupo cíclico $(\mathbb{F}_q \backslash \{0\}, \cdot)$. Definimos $a_1 = 0$, $a_i = \alpha^{i-1}$ para todo $i \in \{1, \ldots, q-1\}$ y L_s el espacio vectorial de polinomios de grado menor o igual que s, $L_s = \{f(x) \in \mathbb{F}_q[x] \mid \deg(f) \leq s\}$ el código Reed-Solomon $RS_q(s)$ es la imagen de la aplicación evaluación:

$$\begin{aligned} \text{ev} : L_s &\longrightarrow \mathbb{F}_q^n \\ f &\longrightarrow (f(a_1), \ldots, f(a_n)) \end{aligned}$$

entonces se define el código Reed-Solomon $\mathcal{C} = RS_q(s)$ de dimensión $k(\mathcal{C}) = s+1$ en $\mathbb{F}_q$ como $RS_q(s) = \{\text{ev}(f) \mid f \in L_s\}$.

El código $(C) = RS_q(s)$ es un código con $n(\mathcal{C}) = q$, $k(\mathcal{C}) = s+1$ y $d(\mathcal{C}) = n(\mathcal{C}) - k(\mathcal{C}) + 1 = q - s$ (sus parámetros fueron estudiados en el Teorema 3.17). Por tanto, se trata de un código MDS, (ver Definición 2.13), con la mejor distancia mínima posible fijados sus parámetros $n(\mathcal{C})$ y $k(\mathcal{C})$. La capacidad de corrección del código Reed-Solomon $RS_q(s)$ es $t_{RS} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor = \lfloor \frac{q-s-1}{2} \rfloor$, que coincide con el número de errores del algoritmo que presentaremos (Algoritmo 4.1).

Nota 5 *Los códigos Reed Solomon se pueden ver como casos particulares de los códigos Reed-Muller y códigos Cubo,* $RS_q(s) = RM_q(s,1) = Cubo_q(s,1)$.

Supongamos que enviamos la palabra $c \in RS_q(s) = \mathcal{C}$. Como $c \in RS_q(s)$, sabemos que existe un polinomio $f \in L_s$ tal que $c = \text{ev}(f)$. Tras enviar la palabra c por un canal, recibimos el vector $y \in \mathbb{F}_q^n$ donde se han producido a lo sumo t errores donde $t = t_{RS}$ es la capacidad de corrección de $RS_q(s)$. Es decir, utilizando la Proposición 2.16 $y = c + e$, con $e \in \mathbb{F}_q^n$ y $w_H(e) \leq \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor = \lfloor \frac{q-s-1}{2} \rfloor$. Dicho de otra forma, buscamos un polinomio $f \in L_s$ tal que $d_H(y, RS_q(s)) = d_H(y, \text{ev}(f))$.

El objetivo de esta sección será el de proporcionar y demostrar la corrección del algoritmo de la Figura 4.1 que nos permitirá encontrar este polinomio $f \in L_s$ a través de la resolución de un sistema de ecuaciones lineales, permitiéndonos de esta forma definir un algoritmo de decodificación eficiente de la familia de códigos Reed-Solomon.

Veamos que, efectivamente, el Algoritmo 4.1, que es un algoritmo de decodificación de $\mathcal{C} = RS_q(s)$ corrige hasta $t = t_{RS} = \lfloor \frac{n(\mathcal{C})-k(\mathcal{C})}{2} \rfloor = \lfloor \frac{q-s-1}{2} \rfloor$ errores. Antes necesitamos un resultado previo.

Algoritmo $\mathrm{Dec}_{RS}(s, y)$

Input : $y \in \mathbb{F}_q^n, s \in \mathbb{N}$.
Output : $c \in \mathcal{C} = RS_q(s)$ *tal que* $d_H(y, c) \leq t = \lfloor \frac{n(\mathcal{C})-k(\mathcal{C})}{2} \rfloor$
o bien Error.

Consideramos los polinomios:

$$E(x) = x^t + \sum_{i=0}^{t-1} A_i x^i \in \mathbb{F}_q[A_0, \ldots, A_{t-1}][x],$$

$$P(x) = \sum_{j=0}^{k(\mathcal{C})+t-1} B_j x^j \in \mathbb{F}_q[B_0, \ldots, B_{k(\mathcal{C})+t-1}][x].$$

Resolvemos el siguiente sistema de ecuaciones lineales en las variables A_i,B_j con $i \in \{0, \ldots, t-1\}$ y $j \in \{0, \ldots, k(\mathcal{C}) + t - 1\}$:

$$\begin{cases} P(a_1) - E(a_1)y_1 & = 0 \\ \quad \vdots & \quad \vdots \\ P(a_n) - E(a_n)y_n & = 0 \end{cases}$$

if El sistema es incompatible **then**
  return Error.
end if
Sea $A_i = \alpha_i$ y $B_j = \beta_j$ una solución del sistema con $i \in \{0, \ldots, t-1\}$ y $j \in \{0, \ldots, k(\mathcal{C}) + t - 1\}$.
$f(x) := \frac{P(x)}{E(x)}$.
if Si $f(x) \notin \mathbb{F}_q[x]$ **then**
  return Error.
end if
if $d_H(y, \mathrm{ev}(f)) > t$ **then**
  return Error.
end if
return $c = \mathrm{ev}(f)$.

Figura 4.1: Pseudocódigo del algoritmo de decodificación de códigos Reed-Solomon.

Lema 4.1. *Sea $y \in \mathbb{F}_q^n$ $\mathcal{C} = RS_q(s)$ tal que $d_H(y, \mathcal{C}) \leq t_{RS}$. Además, sean:*

$$E(x) = x^{t_{RS}} + \sum_{i=0}^{t_{RS}-1} A_i x^i \in \mathbb{F}_q[A_0, \ldots, A_{t_{RS}-1}][x],$$

$$P(x) = \sum_{i=0}^{k(\mathcal{C})+t_{RS}-1} B_i x^i \in \mathbb{F}_q[B_0, \ldots, B_{k(\mathcal{C})+t_{RS}-1}][x].$$

Entonces, el sistema lineal de ecuaciones en las variables A_i y B_j:

$$\begin{cases} P(a_1) - E(a_1)y_1 & = 0 \\ \quad \vdots & \quad \vdots \\ P(a_n) - E(a_n)y_n & = 0 \end{cases} \tag{4.1}$$

es un sistema compatible.

Demostración. Sea $t = t_{RS}$ y supongamos que $d_H(y, \mathcal{C}) \leq t$, entonces existe $g \in L_s$ tal que $d_H(y, \mathrm{ev}(g)) \leq t$. Sea $I = \{i_1, \ldots, i_l\} = \{i \in \{1, \ldots, n\} \mid y_i \neq g(a_i)\}$ el conjunto de posiciones de error, es decir, donde $\mathrm{ev}(g)$ e y no coinciden, se tiene que $l \leq t$. Definimos:

$$E(x) = \prod_{j=1}^{l} (x - a_{i_j}) \cdot x^{t-l} \ , \ P(x) = E(x)g(x).$$

Se observa que $E(x)$ es un polinomio mónico de grado t y $P(x) \in L_{k(\mathcal{C})+t-1}$. Además, $P(a_i) = E(a_i)y_i$ para todo $i \in \{1, \ldots, n\}$ ya que si $i \in I$ entonces $E(a_i) = 0$ y en caso contrario, $g(a_i) = y_i$. Luego, $E(a_i)g(a_i) = E(a_i)y_i$. Por lo tanto, el sistema propuesto tiene solución. □

Teorema 4.2. *El algoritmo $Dec_{RS}(s, v)$ de la Figura 4.1 es un algoritmo de decodificación por mínimas distancias para el código $RS_q(s) = \mathcal{C}$ que corrige $t \leq t_{RS} = \lfloor \frac{n(\mathcal{C})-k(\mathcal{C})}{2} \rfloor$ errores y devuelve "Error"si la distancia de Hamming de la palabra recibida a $\mathcal{C}$ es mayor a t.*

Demostración. Supongamos que la palabra recibida $y \in \mathbb{F}_q^n$ está a una distancia menor o igual a t de $\mathcal{C}$. Sea $E(x) = x^t \sum_{i=0}^{t-1} A_i x^i$ mónico y $P(x) = \sum_{i=0}^{k(\mathcal{C})+t-1} B_i x^i$ una solución no nula del sistema descrito en (4.1), es decir, $P(a_i) - E(a_i)y_i = 0$, para todo $i \in \{1, \ldots, n\}$ (esta solución existe por Lema 4.1). Como $d_H(y, \mathcal{C}) = t$, sabemos que existe $g \in L_s$ tal que $d_H(y, \mathrm{ev}(f)) = d_H(y, \mathcal{C}) = t$.
Sea $I = \{i \in \{1, \ldots, n\} \mid y_i \neq g(a_i)\}$, entonces para todo $i \in \{1, \ldots, n\} \backslash I$ se tiene que $P(a_i) - E(a_i)g(a_i) = P(a_i) - E(a_i)y_i = 0$. Luego, tenemos que $P(x) - E(x)g(x)$ es un polinomio de grado $\deg(P(x) - E(x)g(x)) \leq k(\mathcal{C}) + t - 1$

que tiene al menos $n(\mathcal{C}) - t$ ceros. Como $t \leq \lfloor \frac{n(\mathcal{C})-k(\mathcal{C})}{2} \rfloor < \frac{n(\mathcal{C})-k(\mathcal{C})+1}{2}$, entonces $k(\mathcal{C}) + t - 1 < n(\mathcal{C}) - t$ y tenemos que $P(x) - E(x)g(x) = 0$ para todo $x \in \mathbb{F}_q$. Además:

$$\deg(P(x)) = \deg(E(x)g(x)) = k(\mathcal{C}) + t - 1 < n(\mathcal{C}) = q$$

luego $g(x) = \frac{P(x)}{E(x)} \in \mathbb{F}_q[x]$ es la solución buscada. Por último, es fácil ver que si $d_H(y, \mathcal{C}) > t_{RS}$, entonces la salida del algoritmo es "Error", ya que, en particular, al final del algoritmo se comprueba que la distancia de y a $\text{ev}(f)$ es menor o igual a t_{RS}. □

Ilustremos este algoritmo con un ejemplo detallado.

Ejemplo 4.3. Sea el código $RS_5(3)$, la capacidad correctora del algoritmo coincide con la capacidad correctora del código $RS_5(3)$, $t = \lfloor \frac{5-3}{2} \rfloor = \lfloor \frac{2}{2} \rfloor = 1$. Supongamos que el emisor quiere enviar el vector $(1, 2, 3, 4, 0)$ y que tras enviarse a través del canal se produce un error, de manera que el receptor recibe el vector $y = (1, 2, 3, 4, 1)$. En este caso, sean:

$$E(x) = x + A_0 \in \mathbb{F}_5[A_0][x],$$

$$P(x) = B_3x^3 + B_2x^2 + B_1x + B_0 \in \mathbb{F}_5[B_0, B_1, B_2, B_3][x].$$

Planteamos el sistema de ecuaciones:

$$\begin{cases} P(0) - 1E(0) &= 0 \\ P(1) - 2E(1) &= 0 \\ P(2) - 3E(2) &= 0 \\ P(3) - 4E(3) &= 0 \\ P(4) - 1E(4) &= 0 \end{cases} = \begin{cases} B_0 + 4A_0 &= 0 \\ B_3 + B_2 + B_1 + B_0 + 3A_0 &= 2 \\ 3B_3 + 4B_2 + 2B_1 + B_0 + 2A_0 &= 1 \\ 2B_3 + 4B_2 + 3B_1 + B_0 + A_0 &= 2 \\ 4B_3 + B_2 + 4B_1 + B_0 + 4A_0 &= 4 \end{cases}$$

cuya solución es $(B_3, B_2, B_1, B_0, A_0) = (0, 1, 2, 1, 1)$ A partir de aquí, definimos $f(x) = \frac{x^2+2x+1}{x+1} = x + 1$. Calculamos el vector de evaluaciones de f, $\text{ev}(f) = (1, 2, 3, 4, 0)$ y además, se tiene que $d_H(\text{ev}(f), y) = 1$, luego el algoritmo devuelve $c = \text{ev}(f) \in \mathbb{F}_5^5$ corrigiendo así el error que se cometió.

4.2. Decodificación de códigos Cubo

El objetivo de esta sección es el de proporcionar un algoritmo para decodificar códigos Cubo, el cual corrige a lo sumo $(t_{RS} + 1)^m - 1$ errores, donde $t_{RS} = \lfloor \frac{d(RS_q(s))-1}{2} \rfloor$ es la capacidad de corrección de $RS_q(s)$ (ver Proposición 2.18). Recordemos la definición y los parámetros de los códigos Cubo.

Sea $n = q^m$, a los elementos de $\mathbb{F}_q^n$ los denotamos por a_i con $0 \leq i \leq n$. Si definimos $L_k = \{f(x_1, \ldots, x_m) \in \mathbb{F}_q[x_1, \ldots, x_m] \mid \deg_{x_j}(f) \leq k\}$ y la aplicación evaluación:

$$\begin{aligned} \text{ev} : L_k &\longrightarrow \mathbb{F}_q^n \\ f &\longrightarrow (f(a_1), \ldots, f(a_n)) \end{aligned}$$

entonces $Cubo_q(k,m) = \{\text{ev}(f) \mid f \in L_k\}$.

El código $\mathcal{C} = Cubo_q(k,m)$ es un código con $n(\mathcal{C}) = q^m$, $k(\mathcal{C}) = (t+1)^m$ y $d(\mathcal{C}) = (q-t)^m$ (sus parámetros fueron estudiados en la Proposición 3.22). La capacidad correctora del código $Cubo_q(k,m)$ es $t_{CUBO} = \lfloor \frac{d(Cubo_q(s))-1}{2} \rfloor = \lfloor \frac{(q-k)^m-1}{2} \rfloor$. Sin embargo, vamos a proponer un algoritmo en el que la capacidad de corrección es $t = (t_{RS}+1)^m - 1 = (\frac{q-k-1}{2})^m - 1 = (\frac{q-k+1}{2})^m - 1$, es decir, $t = \frac{t_{CUBO}}{2^{m-1}}$. Este algoritmo (Figura 4.2) será recursivo sobre el valor de m. Además para el caso $m = 1$ este algoritmo coincide con el algoritmo Dec_{RS} (Figura 4.1) ya visto (ver Nota 5).

Teorema 4.4. *El algoritmo $Dec_{CUBO(s,m,v)}$ de la Figura 4.2 es un algoritmo de decodificación por mínimas distancias para el $Cubo_q(s,m)$ que corrige $(t_{RS}+1)^m - 1$ errores, donde $t_{RS} = \lfloor \frac{d(RS_q(s))-1}{2} \rfloor$ es la capacidad de corrección del código $RS_q(s)$. El algoritmo devuelve "Error"si la distancia de Hamming de la palabra recibida al código $Cubo_q(k,m)$ es mayor a $(t_{RS}+1)^m - 1$. Además, este algoritmo hace $(s+1)^m$ llamadas al algoritmo Dec_{RS} (Figura 4.1).*

Demostración. Para facilitar la lectura de la prueba veamos previamente el caso $m = 2$. Nótese que el caso $m = 1$ se corresponde con decodificar un código Reed-Solomon (Nota 5), y esta decodificación fue vista en la Sección 4.1.

Supongamos que $m = 2$, en este caso $n = q^2$. Sea $v \in \mathbb{F}_q^{q^2}$, tal que $v = (v_{11}, \ldots, v_{1q}, v_{21}, \ldots, v_{2q}, \ldots, v_{q1}, \ldots, v_{qq})$ y supongamos que $d_H(v, Cubo_q(k,m)) < (t_{RS}+1)^2$. Separamos este vector en q-uplas, es decir $o_i = (v_{i1}, \ldots, v_{iq})$. Sea $f(x,y) = \sum_{i=0}^{s} \sum_{j=1}^{s} b_{ij} x^i y^j$ el único polinomio tal que, si denotamos $u = \text{ev}(f)$, se tiene que $d_H(u,v) \leq t$. Veamos cómo el algoritmo propuesto recupera el polinomio f.

Definimos $e_i = |\{j \in \{1, \ldots, m\} \mid u_{ij} \neq v_{ij}\}|$, para todo $i \in \{1, \ldots, q\}$, como el número de errores cometidos en cada q-upla. Diremos que la q-upla o_i es buena si $e_i \leq t_{RS}$ y mala en caso contrario. Ahora, tenemos que:

$$(t_{RS}+1)^2 > \sum_{i=1}^{q} e_i \geq \sum_{o_i malos} e_i \geq (t_{RS}+1)E$$

donde E denota el número de q-uplas o_i malas. De la relación anterior se deduce que $E < t_{RS}+1$. Por lo tanto, $\overline{E} = q - E$ que denota el número de q-uplas o_i buenas, verifica de nuevo que $\overline{E} \geq q - t_{RS}$. Por otro lado, sabemos que f se puede escribir como $f = \sum_{j=1}^{s} h_j(x) y^j$, con $h_j \in \mathbb{F}_q[x]$ y $\deg(h_j(x)) \leq s$.

Definimos $g_l(y) = f(a_l, y) = \sum_{j=0}^{s} h_j(a_l) y^j$, para todo $l \in \{1, \ldots, q\}$, luego $g_l(y) \in \mathbb{F}_q[y]$ con $\deg(g_l(y)) \leq s$. Si e_l es bueno, dada la q-upla $o_l = (v_{l1}, \ldots, v_{lq})$,

Algoritmo $\mathrm{Dec}_{Cubo(s,m,v)}$

Input : $v \in \mathbb{F}_q^{q^m}, s, m \in \mathbb{N}$.
Output : $f \in \mathbb{F}_q[x_1, \ldots, x_m]$ *tal que* $d_H(\mathrm{ev}(f), y) \leq (t_{RS}+1)^m$.
o Error.

$v = (v_1, \ldots, v_q)$ con $v_l \in \mathbb{F}_q^{q^{m-1}}$.
Para cada l se aplica $Dec_{Cubo(s,m,v_l)}$, que devuelve Error o

$$g_l(x_2, \ldots, x_m) = \sum_{j_1,\ldots,j_m=0}^{s} a^l_{j_1,\ldots,j_m} x_2^{j_2} \cdots x_m^{j_m}.$$

if El número de veces que devuelve Error supera $t_{RS}+1$ **then**
  return Error.
end if
for $1 \leq j_1, \ldots, j_m \leq s$ **do**
  $u = (u_1, \ldots, u_q)$.
  if Devuelve Error en la l-ésima llamada **then**
    $u_l = 0$.
  end if
  $u_l = a^l_{j_1,\ldots,j_m}$.
end for
Se toma $h_{j_1,\ldots,j_m}(x_1) \in \mathbb{F}_q[x]$ con $\deg(h_{j_1,\ldots,j_m}(x_1)) \leq s$ tal que $\mathrm{ev}(h_{j_1,\ldots,j_m}(x_1)) = Dec_{RS}(s, u_l)$.
if Algún $Dec_{RS}(s, u_l)$ devuelve Error **then**
  return Error.
end if

$$f = \sum_{j_1,\ldots,j_m=0}^{s} h_{j_1,\ldots,j_m}(x_1) x_2^{j_2} \cdots x_m^{j_m}.$$

if $d_H(\mathrm{ev}(f), v) > (t_{RS}+1)^m$ **then**
  return Error.
end if
return

$$f = \sum_{j_1,\ldots,j_m=0}^{s} h_{j_1,\ldots,j_m}(x_1) x_2^{j_2} \cdots x_m^{j_m}.$$

Figura 4.2: Pseudocódigo del algoritmo de decodificación de códigos Cubo. Para simplificar su exposición, la salida del algoritmo es f. Si se quisiera obtener una palabra del código, bastará calcular $\mathrm{ev}(f)$.

al aplicar el decodificador Dec_{RS} (Figura 4.1) obtenemos g_l, si no se han producido más de $(t_{RS}+1)^m$ errores. Entonces, recuperamos $h_j(a_l)$ para cada $j \in \{1,\ldots,s\}$ y habrá al menos $q - t_{RS}$ evaluaciones buenas de $h_j(x)$. Por tanto, aplicamos el decodificador Dec_{RS} (Figura 4.1) y recuperamos el polinomio $h_j(x)$. Si repetimos el proceso $s+1$ veces, obtendremos $f(x,y)$.

Generalicemos el razonamiento anterior para m variables. Sea v un vector $v = (v_1,\ldots,v_q)$ tal que $v_i \in \mathbb{F}_q^{q^{m-1}}$ para todo $i \in \{1,\ldots,q\}$. Por el razonamiento anterior, podemos afirmar la existencia de un polinomio f, que se puede escribir como $f(x_1,\ldots,x_m) = \sum_{j_2\ldots,j_m=0}^{s} h_{j_2,\ldots,j_m}(x_1)x_2^{j_2}\cdots x_m^{j_m}$, con $\deg(h_{j_2,\ldots,j_m}) \leq s$, luego $\mathrm{ev}(h_{j_2,\ldots,j_m}) \in Cubo_q(s, m-1)$.

Definimos $e_i = |\{v_{ij_1\cdots j_{m-1}} \mid v_{ij_1\cdots j_{m-1}} \neq f(a_i, a_{j_1},\ldots,a_{j_{m-1}})\}|$ para todo $i \in \{1,\ldots,q\}$ análogamente a la definición que se dio antes. Ahora, decimos que la q^{m-1}-upla v_i es buena si $e_i < (t_{RS}+1)^{m-1}$ y malo, en caso contrario. Tenemos que:

$$(t_{RS}+1)^m > \sum_{i=1}^{q} e_i \geq \sum_{v_i malos} e_i \geq (t_{RS}+1)^{m-1}E$$

donde E denota el número de v_i malos. De la fórmula anterior, se deduce que $E < (t_{RS}+1)$. Ahora podemos definir:

$$g_l(x_2,\ldots,x_m) = f(a_l,x_2,\ldots,x_m) = \sum_{j_2,\ldots,j_l=0}^{s} h_{j_2,\ldots,j_m}(a_l)x_2^{j_2}\cdots x_m^{j_m}$$

para todo $l \in \{1,\ldots,q\}$, luego $g_l \in \mathbb{F}_q[x_2,\ldots,x_m]$ y $\mathrm{ev}(g_l) \in Cubo_q(s, m-1)$. Si e_l es bueno, aplicamos el decodificador del $Cubo(s, m-1)$ y obtenemos g_l. Este proceso lo haremos para cada $l \in \{1,\ldots,q\}$. Para recuperar cada g_l aplicamos, por tanto, q veces el decodificador $Dec_{CUBO(s,m)}$. Una vez recuperado g_l, tenemos $h_{j_2,\ldots,j_m}(a_l)$, con $0 \leq j_2,\ldots,j_m \leq s$. Queremos ahora recuperar $h_{j_2,\ldots,j_m}(x_1) \in \mathbb{F}_q[x_1]$, con $\deg(h_{j_2,\ldots,j_m}(x_1)) \leq s$, donde sabemos que $h_{j_2,\ldots,j_m}(a_i)$ son correctos para cada i que se corresponda con un v_i bueno. Recordemos que el número de v_i buenos es $q^{m-1} - E \geq q^{m-1} - (t_{RS}+1)$. Luego, para recuperar $h_{j_2,\ldots,j_m}(x_1)$, bastará aplicar el decodificador Dec_{RS} (Figura 4.1). Tendremos que repetir el proceso para cada $j_2,\ldots,j_m \in \{0,\ldots,s\}$ luego debemos aplicar Dec_{RS} (Figura 4.1) $(s+1)^m$ veces para obtener $f(x_1,\ldots,x_m)$. □

4.3. Decodificación de códigos Reed-Muller binarios

El objetivo de esta sección será el de proporcionar un algoritmo para decodificar códigos Reed-Muller binarios, $RM_2(r,m)$, el cual corregirá a lo sumo t_{RM} errores, donde $t_{RM} = 2^{m-r-1} - 1$. Este algoritmo fue introducido por Reed en [17] en el año 1954. Recordemos la definición y los parámetros de los código Reed-Muller, en particular, de los códigos Reed-Muller binarios.

Antes de presentar el algoritmo es conveniente introducir la notación que vamos a usar. Sea $n = 2^m$, para los puntos de $\mathbb{F}_2^m$ vamos a usar la siguiente notación:

Definición 4.5. *Sea $I, J \subseteq \{1, \ldots, m\}$, denotamos:*

$$P_J := (p_j)_{j=1,\ldots,m} \text{ con } p_j = 1 \text{ si } j \in J \text{ y } p_j = 0 \text{ si } j \notin J.$$

Además, en el algoritmo vamos a usar los siguientes monomios:

$$R_I(x_1, \ldots, x_m) := \prod_{i \in I} x_i.$$

Definimos $L_r = \{f(x_1, \ldots, x_m) \in \mathbb{F}_2[x_1, \ldots, x_m] \mid \deg(f) \leq r\}$ y definimos la aplicación evaluación:

$$\begin{aligned} \text{ev} : L_r &\longrightarrow \mathbb{F}_2^n \\ f &\longrightarrow (f(P_J) \mid J \subseteq \{1, \ldots, m\}) \end{aligned}$$

entonces $RM_2(r, m) = \{\text{ev}(f) \mid f \in L_r\}$.

El código $\mathcal{C} = RM_2(r, m)$ es un código con longitud $n(\mathcal{C}) = 2^m$, dimensión $k(\mathcal{C}) = \sum_{i=0}^{r} \binom{m}{i}$ y distancia mínima $d(\mathcal{C}) = 2^{m-r}$ (sus parámetros fueron estudiados en las Proposiciones 3.18 y 3.20). La capacidad correctora del código Reed-Muller, $RM_2(r, m)$, es $t_{RM} = \lfloor \frac{d(\mathcal{C})-1}{2} \rfloor = \lfloor \frac{2^{m-r}-1}{2} \rfloor = 2^{m-r-1} - 1$. Como veremos, el número de errores que corrige el algoritmo $Reed(r, m, y)$ (Figura 4.3) es $t = 2^{m-r-1} - 1$, que coincide con la capacidad de corrección del código Reed-Muller binario $RM_2(r, m)$.
Veamos que, en efecto, el Algoritmo $Reed(r, m, y)$ (Figura 4.3) corrige $2^{m-r-1} - 1$ errores, previamente, veamos algunos lemas que nos serán de utilidad.

Lema 4.6. *Sean $I, J \subseteq \{1, \ldots, m\}$, entonces* $R_I(P_J) = \begin{cases} 1, & \text{si } I \subseteq J \\ 0, & \text{otro caso.} \end{cases}$

Demostración. Directo de la definición de R_I y P_J. □

Lema 4.7. *Sea $I \subseteq \{1, \ldots, m\}$, entonces:*

$$\sum_{J \subseteq I} R_I(P_J) = 1.$$

Demostración. $\sum_{J \subseteq I} R_I(P_J) = \sum_{J \subsetneq I} R_I(P_J) + R_I(P_I)$ y por el Lema 4.6, $\sum_{J \subsetneq I} R_I(P_J) = 0$ y $R_I(P_I) = 1$. Luego tenemos que $\sum_{J \subseteq I} R_I(P_J) = 1$. □

Sea $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ de grado r, la siguiente proposición muestra cómo obtener cada uno de los coeficientes de f de 2^{m-r} formas diferentes, una por cada posible elección del conjunto L descrito en la Proposición 3.20, a partir de los valores de la evaluación de f en los puntos de $\mathbb{F}_2$.

Algoritmo Reed(r, m, y)

Input : $y \in \mathbb{F}_2^{2^m}, r, m \in \mathbb{N}$.
Output : $c \in RM_2(r, m)$ *tal que* $d_H(y, c) \leq t_{RM}$.
o Error.

Para todo $I \subseteq \{1, \ldots, m\}$ tal que $|I| = r$ y $L \cap I = \emptyset$:
Se plantean las ecuaciones $a_I = \sum_{J \subseteq I} y_{J \cup L}$ y cada una devuelve 1 o 0. Se denota por N_0 al número de ceros y por N_1 al número de unos.
if $N_0 = N_1$ o mín$\{N_0, N_1\} > t$ **then**
  return Error.
end if
Definimos $a_I = \begin{cases} 0, & \text{si } N_0 > N_1. \\ 1, & \text{si } N_1 > N_0. \end{cases}$ y $f_r = \sum_{|I|=r} a_I R_I$.
Sea $y = y - ev(f_r)$.
if $r = 0$ **then**
  return ev(f_r).
end if
return $c = \text{ev}(f_r) + Reed(m, r-1, y)$.

Figura 4.3: Pseudocódigo del algoritmo de decodificación de códigos Reed-Muller.

Proposición 4.8. *Sean*

$$f = \sum_{\substack{I \subseteq \{1,\ldots,m\} \\ |I| \leq r}} a_I R_I(x_1, \ldots, x_m),$$

$\text{ev}(f) = (y_J \mid J \subseteq \{1, \ldots, m\})$. *Si* $|I| = r$ *y* $L \cap I = \emptyset$ *entonces* $a_I = \sum_{J \subseteq I} y_{J \cup L}$.

Demostración.

$$a_I = \sum_{J \subseteq I} \alpha_{J \cup L} = \sum_{J \subseteq I} \sum_{|K| \leq r} a_K R_K(P_{J \cup L}) = \sum_{|K| \leq r} (\sum_{J \subseteq I} R_K(P_{J \cup L})) a_K.$$

Ahora, si $K = I$, $\sum_{J \subseteq I} R_I(P_{J \cup L})$, en caso contrario, si $K \neq I$ existe $i \in I \backslash K$ tal que:

$$\sum_{J \subseteq I} R_K(P_{J \cup L}) = \sum_{\substack{J \subseteq I \\ i \in J}} R_K(P_{J \cup L}) + \sum_{\substack{J \subseteq I \\ i \notin J}} R_K(P_{J \cup L}) = 2 \sum_{\substack{J \subseteq I \\ i \in J}} R_K(P_{J \cup L}) = 0.$$

Luego $a_I = \sum_{|K| \leq r} a_K = \text{ev}(f) = y_J = y_{J \cup L}$. □

Cuando la palabra es del código, podemos recuperar cada coeficiente del polinomio de 2^{m-r} formas. Además, los conjuntos de entradas involucrados en cada una de estas formas son disjuntos. Ilustremos esto con un ejemplo.

Ejemplo 4.9. Sea el código $RM_2(2,4)$ donde:

$$L_2 = \{1, x_1, x_2, x_3, x_4, x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4\}.$$

Por otro lado, sea $f = a_\emptyset + a_{\{1\}}x_1 + a_{\{2\}}x_2 + \cdots + a_{\{3,4\}}x_3x_4$. Evaluando cada punto P_J en los distintos R_I, tenemos (ver Tabla 4.1):

	$P_\emptyset$	$P_{\{1\}}$	$P_{\{2\}}$	$P_{\{3\}}$	$P_{\{4\}}$	$P_{\{1,2\}}$	$P_{\{1,3\}}$	$P_{\{1,4\}}$	$P_{\{2,3\}}$	$P_{\{2,4\}}$	$P_{\{3,4\}}$	$P_{\{1,2,3\}}$	$P_{\{1,2,4\}}$	$P_{\{1,3,4\}}$	$P_{\{2,3,4\}}$	$P_{\{1,2,3,4\}}$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
x_2	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1
x_3	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1
x_4	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1
x_1x_2	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1
x_1x_3	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1
x_1x_4	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1
x_2x_3	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1
x_2x_4	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1
x_3x_4	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1

Tabla 4.1

Tomamos el vector formado por los coeficientes de f, $(a_\emptyset, a_{\{1\}}, \ldots, a_{\{3,4\}})$ y lo multiplicamos por la matriz que define la tabla (Tabla 4.1), al resultado lo denotamos por $(v_1, \ldots, v_{16})$. Entonces por la Proposición 4.8 tenemos que:

$$\begin{aligned} a_{\{1,2\}} &= v_1 + v_2 + v_3 + v_6 \\ &= v_4 + v_7 + v_9 + v_{12} \\ &= v_5 + v_8 + v_{10} + v_{13} \\ &= v_{11} + v_{14} + v_{15} + v_{16} \end{aligned}$$

Las ecuaciones anteriores no tienen elementos v_j en común, luego si se producen errores (el máximo número de errores que se pueden producir son a lo sumo t_{RM}), la mayoría de las ecuaciones anteriores seguirán siendo correctas, permitiéndonos así calcular $a_{\{1,2\}}$. Este razonamiento será análogo para el resto de los coeficientes de los términos de grado 2 de f. Obsérvese, que como se vio en la Proposición 4.8, hay $2^{4-2} = 4$ posibilidades para calcular los coeficientes. Para el resto de coeficientes se tiene:

$$\begin{aligned} a_{\{1,3\}} &= v_1 + v_2 + v_4 + v_7 \\ &= v_3 + v_6 + v_9 + v_{12} \\ &= v_5 + v_8 + v_{11} + v_{14} \\ &= v_{10} + v_{13} + v_{15} + v_{16} \end{aligned} \qquad \begin{aligned} a_{\{1,4\}} &= v_1 + v_2 + v_5 + v_8 \\ &= v_3 + v_6 + v_{10} + v_{13} \\ &= v_4 + v_7 + v_{11} + v_{14} \\ &= v_9 + v_{12} + v_{15} + v_{16} \end{aligned}$$

$$\begin{aligned} a_{\{2,3\}} &= v_1 + v_3 + v_4 + v_9 \\ &= v_2 + v_6 + v_7 + v_{12} \\ &= v_5 + v_{10} + v_{11} + v_{15} \\ &= v_8 + v_{13} + v_{14} + v_{16} \end{aligned} \qquad \begin{aligned} a_{\{2,4\}} &= v_1 + v_3 + v_5 + v_{10} \\ &= v_2 + v_6 + v_8 + v_{13} \\ &= v_4 + v_9 + v_{11} + v_{15} \\ &= v_7 + v_{12} + v_{14} + v_{16} \end{aligned}$$

$$\begin{aligned} a_{\{3,4\}} &= v_1 + v_4 + v_5 + v_{11} \\ &= v_2 + v_7 + v_8 + v_{14} \\ &= v_3 + v_9 + v_{10} + v_{15} \\ &= v_6 + v_{12} + v_{13} + v_{16} \end{aligned}$$

Ahora tenemos que $f = a_\emptyset + a_{\{1\}}x_1 + a_{\{2\}}x_2 + a_{\{3\}}x_3 + a_{\{4\}}x_4 + g$, donde g es el polinomio cuyos términos son los términos de grado 2 de f. Como $\mathrm{ev}(f-g) = \mathrm{ev}(f) - \mathrm{ev}(g)$, aplicamos este proceso a $f - g$ y obtendremos los coeficientes de los términos de f de grado 1. Evaluando de la misma forma se tiene que, para estos coeficientes:

	$P_\emptyset$	$P_{\{1\}}$	$P_{\{2\}}$	$P_{\{3\}}$	$P_{\{4\}}$	$P_{\{1,2\}}$	$P_{\{1,3\}}$	$P_{\{1,4\}}$	$P_{\{2,3\}}$	$P_{\{2,4\}}$	$P_{\{3,4\}}$	$P_{\{1,2,3\}}$	$P_{\{1,2,4\}}$	$P_{\{1,3,4\}}$	$P_{\{2,3,4\}}$	$P_{\{1,2,3,4\}}$
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
x_1	0	1	0	0	0	1	1	1	0	0	0	1	1	1	0	1
x_2	0	0	1	0	0	1	0	0	1	1	0	1	1	0	1	1
x_3	0	0	0	1	0	0	1	0	1	0	1	1	0	1	1	1
x_4	0	0	0	0	1	0	0	1	0	1	1	0	1	1	1	1

Tabla 4.2

$$\begin{aligned} a_{\{1\}} &= v_1 + v_2 \\ &= v_3 + v_6 \\ &= v_4 + v_7 \\ &= v_5 + v_8 \\ &= v_9 + v_{12} \\ &= v_{10} + v_{13} \\ &= v_{11} + v_{14} \\ &= v_{15} + v_{15} \end{aligned} \qquad \begin{aligned} a_{\{2\}} &= v_1 + v_3 \\ &= v_2 + v_6 \\ &= v_4 + v_9 \\ &= v_5 + v_{10} \\ &= v_7 + v_{12} \\ &= v_8 + v_{13} \\ &= v_{11} + v_{15} \\ &= v_{14} + v_{16} \end{aligned} \qquad \begin{aligned} a_{\{3\}} &= v_1 + v_4 \\ &= v_2 + v_7 \\ &= v_3 + v_9 \\ &= v_5 + v_{11} \\ &= v_6 + v_{12} \\ &= v_8 + v_{14} \\ &= v_{10} + v_{15} \\ &= v_{13} + v_{16} \end{aligned} \qquad \begin{aligned} a_{\{4\}} &= v_1 + v_5 \\ &= v_2 + v_8 \\ &= v_3 + v_{10} \\ &= v_4 + v_{11} \\ &= v_6 + v_{13} \\ &= v_7 + v_{14} \\ &= v_9 + v_{15} \\ &= v_{12} + v_{16} \end{aligned}$$

Tabla 4.3

De esta manera ya hemos obtenido los coeficientes de los términos de grado 1 y 2 del polinomio f, si ahora definimos como h al polinomio cuyos términos son los términos de grado 1 y 2 de f, de manera similar a antes, tenemos que $\text{ev}(f-h) = \text{ev}(f) - \text{ev}(h)$, luego aplicamos de nuevo este proceso a $f-h$ y obtendremos el término $a_\emptyset$. Por lo tanto, obtenemos el polinomio f.

Teorema 4.10. *El algoritmo Reed(r, m, y) de la Figura 4.3 es un algoritmo de decodificación para el código $RM_2(r, m)$ que corrige $2^{m-r-1} - 1$ errores.*

Demostración. En consecuencia de la Proposición 4.8, si el número de errores es menor o igual que $2^{m-r-1} - 1$, habrá más formas correctas para obtener a_I que incorrectas, luego aplicando la mayoría de voto[1] obtendríamos el valor correcto de a_I.

Como las entradas de $\mathbf{y}$ son disjuntas, los errores de la i-ésima iteración del método no se transmiten a la $(i+1)$-ésima iteración. Esto se debe a que en cada iteración calculamos los términos de grado $r-i$ (iteración i-ésima) del polinomio f, así que al definir el polinomio con los términos encontrados, g, y restar las evaluaciones de f y g, los términos de grado inferior a $r-i$ no se verán afectados imposibilitando que se transmitan errores. □

Ilustremos este algoritmo con un ejemplo detallado.

Ejemplo 4.11. Sea el código $RM_2(2, 4)$, supongamos que el emisor nos quiere enviar un mensaje a partir del polinomio:

$$f(x_1, x_2, x_3, x_4) = 1 + x_1 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_3x_4.$$

El vector de coeficientes en este caso será $(1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1)$. Por tanto, si consideramos la matriz A formada a partir de las entradas de la tabla 4.1, tenemos:

$$(v_1, \ldots, v_{16}) = (1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1) \cdot A =$$
$$= (1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1).$$

Supongamos que, al querer transmitir este mensaje se comete un error en la entrada v_1 (dato que el receptor desconoce), por lo que el receptor recibe la palabra $\mathbf{y} = (0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1)$. Calculemos los coeficientes de f usando las ecuaciones que previamente hemos presentado, la fórmula que incluye al y_1 estará siempre mal ya que ahí se produjo el error, aunque el receptor lo desconozca. Sin embargo esto no supondrá un problema debido a que las entradas y_i son disjuntas. Para resolver este problema se acudirá a un método denominado *mayoría de voto*, es decir, al resolver estas ecuaciones el dato que más se repita será el correcto. Tiene coherencia aplicar este método ya que en este caso se han producido errores dentro de la capacidad correctora del algoritmo.

[1] Si tenemos varios resultados posibles para una incógnita, el que más se repita será considerado el correcto, este proceso se llama mayoría de voto.

$$\begin{aligned} a_{\{1,2\}} &= y_1 + y_2 + y_3 + y_6 = 0 \\ &= y_4 + y_7 + y_9 + y_{12} = 1 \\ &= y_5 + y_8 + y_{10} + y_{13} = 1 \\ &= y_{11} + y_{14} + y_{15} + y_{16} = 1 \end{aligned} \qquad \begin{aligned} a_{\{1,3\}} &= y_1 + y_2 + y_4 + y_7 = 0 \\ &= y_3 + y_6 + y_9 + y_{12} = 1 \\ &= y_5 + v_8 + y_{11} + y_{14} = 1 \\ &= y_{10} + y_{13} + y_{15} + y_{16} = 1 \end{aligned}$$

$$\begin{aligned} a_{\{1,4\}} &= y_1 + y_2 + y_5 + y_8 = 1 \\ &= y_3 + y_6 + y_{10} + y_{13} = 0 \\ &= y_4 + y_7 + y_{11} + y_{14} = 0 \\ &= y_9 + y_{12} + y_{15} + y_{16} = 0 \end{aligned} \qquad \begin{aligned} a_{\{2,3\}} &= y_1 + y_3 + y_4 + y_9 = 1 \\ &= y_2 + y_6 + y_7 + y_{12} = 0 \\ &= y_5 + v_{10} + y_{11} + y_{15} = 0 \\ &= y_8 + y_{13} + y_{14} + y_{16} = 0 \end{aligned}$$

$$\begin{aligned} a_{\{2,4\}} &= y_1 + y_3 + y_5 + y_{10} = 1 \\ &= y_2 + y_6 + y_8 + y_{13} = 0 \\ &= y_4 + y_9 + y_{11} + y_{15} = 0 \\ &= y_7 + y_{12} + y_{14} + y_{16} = 0 \end{aligned} \qquad \begin{aligned} a_{\{3,4\}} &= y_1 + y_4 + y_5 + y_{11} = 0 \\ &= y_2 + y_7 + y_8 + y_{14} = 1 \\ &= y_3 + v_9 + y_{10} + y_{15} = 1 \\ &= y_6 + y_{12} + y_{13} + y_{16} = 1 \end{aligned}$$

Por mayoría, se tiene que los coeficientes son $a_{\{1,2\}} = 1$, $a_{\{1,3\}} = 1$, $a_{\{1,4\}} = 0$, $a_{\{2,3\}} = 0$, $a_{\{2,4\}} = 0$ y $a_{\{3,4\}} = 1$. Tenemos ahora, si llamamos $g = x_1x_2 + x_1x_3 + x_3x_4$, $\mathrm{ev}(f - g) = \mathrm{ev}(f) - \mathrm{ev}(g)$, luego $y = y - \mathrm{ev}(g) = (0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0)$. Aplicando de nuevo el proceso a $f-g$, donde y ya lo hemos definido y la matriz A la formada a partir de las entradas de la Tabla 4.2, tenemos que:

$$\begin{aligned} a_{\{1\}} &= y_1 + y_2 = 0 \\ &= y_3 + y_6 = 1 \\ &= y_4 + y_7 = 1 \\ &= y_5 + y_8 = 1 \\ &= y_9 + y_{12} = 1 \\ &= y_{10} + y_{13} = 1 \\ &= y_{11} + y_{14} = 1 \\ &= y_{15} + y_{15} = 1 \end{aligned} \qquad \begin{aligned} a_{\{2\}} &= y_1 + y_3 = 1 \\ &= y_2 + y_6 = 0 \\ &= y_4 + y_9 = 0 \\ &= y_5 + y_{10} = 0 \\ &= y_7 + y_{12} = 0 \\ &= y_8 + y_{13} = 0 \\ &= y_{11} + y_{15} = 0 \\ &= y_{14} + y_{16} = 0 \end{aligned} \qquad \begin{aligned} a_{\{3\}} &= y_1 + y_4 = 0 \\ &= y_2 + y_7 = 1 \\ &= y_3 + y_9 = 1 \\ &= y_5 + y_{11} = 1 \\ &= y_6 + y_{12} = 1 \\ &= y_8 + y_{14} = 1 \\ &= y_{10} + y_{15} = 1 \\ &= y_{13} + y_{16} = 1 \end{aligned} \qquad \begin{aligned} a_{\{4\}} &= y_1 + y_5 = 0 \\ &= y_2 + y_8 = 1 \\ &= y_3 + y_{10} = 1 \\ &= y_4 + y_{11} = 1 \\ &= y_6 + y_{13} = 1 \\ &= y_7 + y_{14} = 1 \\ &= y_9 + y_{15} = 1 \\ &= y_{12} + y_{16} = 1 \end{aligned}$$

Por mayoría, se tiene que los coeficientes son $a_{\{1\}} = 1$, $a_{\{2\}} = 0$, $a_{\{3\}} = 1$ y $a_{\{4\}} = 1$. Tenemos ahora, si llamamos $h = x_1 + x_3 + x_4 + g$, $\mathrm{ev}(f - h) = \mathrm{ev}(f) - \mathrm{ev}(h)$, luego $y = y - \mathrm{ev}(h) = (0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1)$. Aplicando de nuevo el proceso a $f - h$, tenemos que el $a_\emptyset = 1$ por mayoría. Por tanto se recuperaría el polinomio:

$$f = 1 + x_1 + x_3 + x_4 + x_1x_2 + x_1x_3 + x_3x_4.$$

Bibliografía

[1] E. Berlekamp and L. Welch. Error Correction of Algebraic Block Codes. US Patent 4.633.470 (1968).

[2] E.R. Berlekamp, R.J. McEliece and H.C.A van Tilborg. On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory IT:24 (1978), 384–386.

[3] E. Camps, H. H. López, L.M. Gretchen and E. Sarmiento. Monomial-Cartesian codes under divisibility. Finite Fields and their Applications (2019), 199–204.

[4] S. K. Chebolu and J. Minác. Counting irreducible polynomials over finite fields using the inclusion-exclusion principle. Mathematics magazine 84 (2011), 369–370.

[5] D. Cox, J. Little and D.O'Shea. Ideals, Varieties, and Algorithms 4th ed., Springer-Verlag, 2015.

[6] J. Fitzgerald and R.F. Lax. Decoding affine variety codes using Gröbner bases. Designs, Codes and Cryptography 13 (1998), 147–158.

[7] O.Geil and T.Tøholdt. Footprints or generalized Bezout's theorem. Transactions of the IRE Professional Group on Information Theory 46:2 (2000), 635–641.

[8] O.Geil and T.Tøholdt. On hyperbolic codes. Applied algebra, algebraic algorithms and error-correcting codes. Lecture Notes in Computer Science 2227 (2001), 159–171.

[9] W.C. Huffman and V. Pless. Fundamental of error conecting codes. Library of Congress Cataloguing in Publication data, 2003.

[10] J. Justesen and T.Tøholdt. A course in error conecting codes. European Mathematical Society, 2004.

[11] O.Geil. On the second weight of generalized Reed-Muller codes. Designs, Codes and Cryptography 48 (2008), 323–330.

[12] D. Knuth. The Art of Computer Programming. Positional Number Systems 3rd ed., Addison-Wesley, 1994.

[13] S. Kopparty, N. Ron-Zewi, S. Saraf and M. Wootters. Improved Decoding of Folded Reed-Solomon and Multiplicity Codes. Annual Symposium on Foundations of Computer Science 59 (2018), 212–223.

[14] D.E. Muller. Application of Boolean algebra to switching circuit design and to error detection. Transactions of the I.R.E. Professional Group on Electronic Computers 3 (1954), 6–12.

[15] F. Parvaresh, M. El-Khamy, M. Stepanov, D. Augot, R.J. McEliece and A. Vardy. Algebraic List Decoding of Reed-Solomon product codes. Proceedings of the Tenth International Workshop on Algebraic and Combinatorial Coding Theory (2006), 210–213.

[16] R. Pellikaan, X.-W. Wu, S. Bulygin and R. Jurrius. Codes, Cryptology and Curves with Computer Algebra, Clays, 2017.

[17] I.S. Reed. A class of multiple-error-correcting codes and the decoding scheme. Transactions of the IRE Professional Group on Information Theory 4:4 (1954), 38–49.

[18] I.S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. Journal of Society for Industrial and Applied Mathematics 8:2 (1960), 300–304.

[19] T.Tøholdt, J.M van Lint and R. Pellikaan. Algebraic Geometry Codes, Handbook of coding theory I, II , North-Holland, 1998.

[20] C.H. Wagner. Simpson's Paradox in Real Life. The American Statistician 36 (1982), 46–48.

Parameters and efficient decoding of affine codes

Sección de Matemáticas
Universidad de La Laguna

Alejandro García García

Facultad de Ciencias · Sección de Matemáticas
Universidad de La Laguna
alu0101107202@ull.edu.es

Abstract

The main goal of Coding Theory is to efficient transfer reliable information through a channel with noise. The source will change the original message by a process called encoding, adding redundant information. Once the encoded message is recieved, it starts the most difficult problem, decoding. The goal of this process is to obtain an estimate of the original message from the received message. This theory looks for families of codes that allow detecting and correcting as many errors as posible and this family should have efficient encoding and decoding procedures. In this dissertation, we study the family of affine codes.

1. Linear Codes

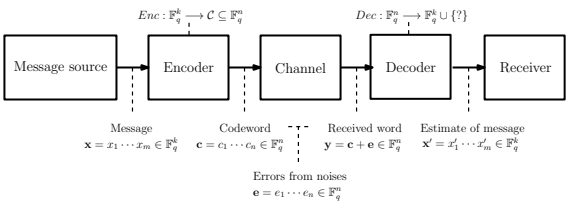


Figure 1: *Communication with Error Correcting Codes.*

Definition 1 *A linear code $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$, its parameters are:*

- *Lenght:* $n(\mathcal{C}) = n$.
- *Dimension:* $k(\mathcal{C}) = \dim(\mathcal{C})$.
- *Minimum distance:* $d(\mathcal{C}) = \min\{|\mathrm{supp}(c)| \mid c \in \mathcal{C}, c \neq 0\}$.

Definition 2 *Let $\mathcal{C}$ a linear code, a decoder by minimum distance that corrects s errors is a map: $Dec : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k \cup \{?\}$ such that if $Enc : \mathbb{F}_q^k \longrightarrow \mathcal{C} \subseteq \mathbb{F}_q^n$ is an encoder for the code $\mathcal{C}$, then $Dec(Enc(m + e)) = m$, for all $e \in \mathbb{F}_q^n$ with $|\mathrm{supp}(e)| \leq s$.*

One can always design a decoder that corrects up to $\lfloor \frac{d(\mathcal{C})-1}{2} \rfloor$ by performing exhaustive search, but this algorithm quickly becomes cumbersome as k increases. Finding efficient decoding algorithms for linear codes is a hard problem.

2. Affine codes

Definition 3 *Let $L \subseteq B = \{x_1^{j_1} \cdots x_m^{j_m} + I \mid 0 \leq j_1, \ldots, j_m < q\}$, $L \neq \emptyset$, we define the affine code $\mathcal{C}_L$ as the image of L under the evaluation map: $\mathrm{ev} : \langle L \rangle \longrightarrow \mathbb{F}_q^n$ defined as $f + I \mapsto (f(P_1), \ldots, f(P_n))$, where $\mathbb{F}_q^n = \{P_1, \ldots, P_n\}$.*

Definition 4 *Let $L \subseteq B$. We define the footprint-bound of the affine code $\mathcal{C}_L$ as the integer:*

$$\mathrm{FB}(\mathcal{C}_L) = \min\{(q - \beta_1) \cdots (q - \beta_m) \mid x_1^{\beta_1} \cdots x_m^{\beta_m} \in L\}.$$

Theorem 1 *Let $L \subseteq B$. Then $n(\mathcal{C}_L) = q^m$, $k(\mathcal{C}_L) = |L|$ and $d(\mathcal{C}_L) \geq \mathrm{FB}(\mathcal{C}_L)$.*

Some remarkable families of codes are the Reed-Solomon, the Reed-Muller, the Cubes and the Hyperbolics.

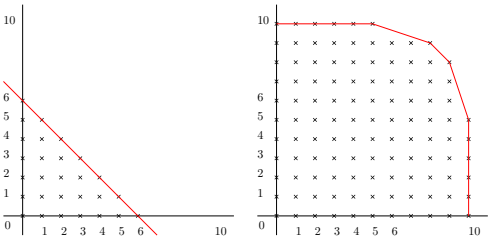


Figure 2: *If $L_1 = \{x_1^{\alpha_1} x_2^{\alpha_2} \mid 0 \leq \alpha_1, \alpha_2 \leq 6, \alpha_1 + \alpha_2 \leq 6\}$ then $\mathcal{C}_{L_1}$ is a Reed-Muller code. If $L_2 = \{x_1^{\alpha_1} x_2^{\alpha_2} \mid 0 \leq \alpha_1, \alpha_2 < 11, (q - \alpha_1) \cdots (q - \alpha_m) \geq 6\}$ then $\mathcal{C}_{L_2}$ is an Hyperbolic code.*

3. A running example: binary Reed-Muller codes

We denote $\mathbb{F}_2^m = \{p_I \mid I \subseteq \{1, \ldots, m\}\}$, where $(p_I)_i = 1$ if $i \in I$ or $(p_I)_i = 0$ otherwise.

Definition 5 *Let $s, m \in \mathbb{Z}^+$ and*

$$L_{RM} = \{f \in \mathbb{F}_2[x_1, \ldots, x_m] \mid \deg(f) \leq s\}.$$

Then the affine code $C_{L_{RM}}$ is called binary Reed-Muller code of degree s in m variables and it is denoted by $RM_2(s, m)$.

Theorem 2 *Let $RM_2(s, m)$ be a binary Reed-Muller code. Then its parameters are: $n(RM_2(s, m)) = 2^m$, $k(RM_2(s, m)) = \sum_{i=0}^{s} \binom{m}{i}$ and $d(RM_2(s, m)) = 2^{m-s}$.*

The following algorithm is due to Reed [1].

Algorithm $\mathrm{Reed}(s, m, y)$

Input : $y \in \mathbb{F}_2^{2^m}, s, m \in \mathbb{N}$.
Output : $c \in RM_2(s, m)$ such that $|\mathrm{supp}(y - c)| \leq 2^{m-s-1} - 1$.
or Error.

For all $I \subseteq \{1, \ldots, m\}$ such that $|I| = r$ and $L \cap I = \emptyset$:
We define the equations $a_I = \sum_{J \subseteq I} y_{J \cup L}$ whose results could be 1 or 0. We denote by N_0 the number of zeros and by N_1 the number of ones.
if $N_0 = N_1$ or $\min\{N_0, N_1\} > 2^{m-s-1} - 1$ **then**
  return Error.
end if
Set $a_I = \begin{cases} 0, & \text{if } N_0 > N_1. \\ 1, & \text{if } N_1 > N_0. \end{cases}$ and $f_s = \sum_{|I|=s} a_I R_I$.
$y = y - ev(f_s)$.
if $s = 0$ **then**
  return $\mathrm{ev}(f_s)$.
end if
return $c = \mathrm{ev}(f_s) + Reed(m, s - 1, y)$.

Figure 3: *Pseudocode of the algorithm to decode Reed-Muller codes.*

Theorem 3 *The algorithm $Reed(s, m, y)$ described in Figure 3 is a decoder for the $RM_2(s, m)$ code that corrects $2^{m-s-1} - 1$ errors.*

References

[1] I.S. Reed. A class of multiple-error-correcting codes and the decoding scheme. Transactions of the IRE Professional Group on Information Theory 4:4 (1954), 38–49.

TRABAJO FIN DE GRADO, Convocatoria de Junio, 2021