Universidad de La Laguna

Sección de Matemáticas | Facultad de Ciencias

Cathaysa Pérez Quintero

Decodificación algebraica de códigos cíclicos

Dado cualquier código cíclico, ¿es fácil corregir errores?

Trabajo Fin de Grado
Grado en Matemáticas
La Laguna, Julio de 2017

Dirigido por

Irene Márquez Corbella
Margarita Rivero Álvarez

Irene Márquez Corbella
Departamento de Matemáticas, Estadística e Investigación Operativa
Universidad de La Laguna
38271 La Laguna, Tenerife

Margarita Rivero Álvarez
Departamento de Matemáticas, Estadística e Investigación Operativa
Universidad de La Laguna
38271 La Laguna, Tenerife

Agradecimientos

Gracias a mis tutoras,
Irene Márquez Corbella y Margarita Rivero Alvárez,
por guiarme y aconsejarme durante este proceso.
A mis profesores del Grado,
por dotarme de las aptitudes adquiridas.
A mi familia,
por el cariño y las palabras de ánimo.
A mis compañeros,
por las horas de estudio compartidas.
Y a Raúl,
por el apoyo y la paciencia.

Resumen · Abstract

Resumen

El objetivo de la teoría de códigos es transferir un mensaje entre emisor y receptor de forma eficiente y fiable. Se necesitan códigos que tengan una tasa de información alta, algoritmos que posean una gran capacidad correctora de errores y que la complejidad del método, tanto en la codificación como en la decodificación, sea baja. Un ejemplo de algoritmo de decodificación son aquellos que utilizan pares correctores de errores (ECP). En este trabajo fin de grado caracterizaremos ECP para códigos cíclicos. En particular, para códigos BCH y Reed-Solomon generalizados (GRS).

Palabras clave: *Pares correctores de errores – Códigos cíclicos – Códigos BCH – Códigos Reed Solomon – Algoritmo de decodificación.*

Abstract

The main goal of coding theory is to efficiently transfer reliable information between sender and receiver. We need codes with high information rate, the algorithm used should have a high error correction capacity and the complexity of coding and decoding should be low. An example of decoding algorithms are those that use error-correcting pairs (ECP). In this thesis degree we will characterize ECP for cyclic codes, in particular for BCH and generalized Reed-Solomon (GRS) codes.

Keywords: *Error correcting pairs – Cyclic codes – BCH codes – Reed-Solomon codes – Decoding algorithm.*

Contenido

Introducción

El objetivo de la teoría de códigos es transferir un mensaje entre emisor y receptor de forma eficiente y fiable. Debemos asegurarnos de que los mensajes enviado y recibido coincidan, a pesar de que hayan podido ser dañados mediante canales con ruido. Por tanto, el problema es encontrar una técnica óptima que permita corregir los errores de un mensaje de forma segura. La idea es añadir información redundante al mensaje para poder corregirlo tras el envío. Para ello, necesitamos códigos con una tasa de información alta, es decir, que la cantidad de información redundante no sea demasiado elevada frente a la información del mensaje; que la capacidad correctora de errores sea alta, esto es, que la técnica sea capaz de corregir la mayor cantidad de posibles errores; y, por último, que la complejidad del método sea baja, tanto en la codificación como en la decodificación.

Una de estas técnicas es el algoritmo de decodificación a partir de "pares correctores de errores", introducido independientemente por Pellikaan [4] y Kötter [5]. Pellikaan nos presenta un algoritmo tal que, dado un código con un par t-corrector de errores, es capaz de corregir hasta t errores con complejidad polinomial. Sin embargo, la dificultad de la decodificación con pares correctores de errores es encontrar el par de códigos que nos permita ejecutarlo.

Este trabajo está organizado en tres capítulos. En el primer capítulo, introduciremos los conceptos básicos sobre la teoría de códigos lineales. En particular, hablaremos de su presentación a partir de las matrices generatriz y de paridad, así como los conceptos de código dual, distancia de Hamming y peso de Hamming. Además explicaremos en qué se basan los métodos de codificación y decodificación con códigos lineales y presentaremos uno de los algoritmos de decodificación para códigos lineales más usados en la literatura la "decodificación por Síndrome".

En el segundo capítulo, nos centraremos en explicar una subfamilia de códigos lineales particular, los códigos cíclicos. Caracterizaremos estos códigos

como ideales de un anillo cociente y presentaremos el concepto de polinomio generador y de control de estos códigos. Por otra parte, definiremos los códigos cíclicos a partir del conjunto de ceros de un polinomio, lo que nos permite definir algunas familias particulares como los códigos BCH, Reed Solomon y Reed Solomon generalizados. De estos últimos presentaremos un algoritmo de decodificación. Durante el primer y segundo capítulo utilizaremos el software libre Sagemath para crear ejemplos, a modo de breve tutorial de su uso aplicado a códigos lineales y cíclicos.

En el tercer y último capítulo, presentaremos el concepto de par corrector de errores (ECP) y se explicará cómo con este concepto se puede definir un algoritmo de decodificación eficiente para cualquier código lineal. Posteriormente, expondremos algunas propiedades interesantes que pueden facilitarnos la búsqueda de ECP. Tras esto, hablaremos de las funciones localizadoras de errores y de cómo se pueden debilitar las condiciones de los pares localizadores de errores. Por último, explicaremos las pautas para construir pares correctores de errores para los códigos Reed-Solomon generalizados y los códigos BCH.

Si la construcción de pares de códigos correctores fuese efectiva para cualquier código lineal, entonces estos resultados tendrían aplicaciones en criptografía de clave pública. La noción de *criptografía de clave pública* fue introducida en 1976 por Diffie-Hellman [10]. La principal ventaja con respecto a la *criptografía de clave simétrica* es que no necesita un intercambio inicial de claves entre emisor y receptor. Matemáticamente, la criptografía de clave pública busca una función que al evaluarla sea sencilla pero que, a su vez, invertir dicha función tenga una complejidad elevada. Los ejemplos más comunes de este tipo de funciones son: la factorización de enteros, el logaritmo discreto y el logaritmo discreto en curvas elípticas. Sin embargo, el algoritmo de Shor y la posible aparición de un ordenador cuántico pone en peligro la criptografía basada en los problemas antes citados. La criptografía basada en códigos correctores, presentada en 1978 por McEliece [8], es un candidato interesante para la criptografía post-cuántica, es decir, la nueva generación de criptografía que resiste la aparición del ordenador cuántico. Se trata de criptosistemas que ofrecen métodos de cifrado y descifrado rápidos pero que tienen el inconveniente de trabajar con claves públicas muy grandes. La seguridad del criptosistema de McEliece está basada en la dificultad del problema de decodificación general de códigos lineales, que fue probado ser un problema NP completo en [9] por Berlekamp, McEliece y van Tilborg.

Se sabe de la existencia de pares correctores de códigos para ciertas familias cómo los códigos Reed-Solomon, BCH, códigos geométricos-algebraicos y códigos Goppa. En este trabajo no sólo se tratará de su existencia sino de una construcción efectiva para algunas familias de códigos. La generalización de los resultados analizados en este trabajo a más familias de códigos tendría, por tanto, graves consecuencias en los criptosistemas basados en códigos.

1

Teoría de códigos

1.1. Introducción a la teoría de la información

En 1948 Claude Shannon[*] publica el artículo *"A mathematical theory of communication"*, que supone el comienzo de la *teoría de la información* y la *teoría de códigos*. La teoría de códigos tiene como objetivo poder transferir un mensaje de forma eficiente y fiable. Para que sea fiable se requiere que el mensaje enviado y recibido coincidan; para que sea eficiente se necesita que no requiera gran cantidad de esfuerzo y tiempo. Sin embargo, la transferencia a través de canales ruidosos puede dañar la información enviada. Por ello, necesitamos detectar si se han producido errores y saber cómo poder corregirlos.

Por otro lado, la Teoría de la Información está relacionada con la capacidad de un canal. Dado un canal de comunicación, Shannon identificó la capacidad máxima de información que puede transportar dicho canal de forma fiable, llamada capacidad del canal, es decir con una probabilidad de error tan pequeña como se quiera.

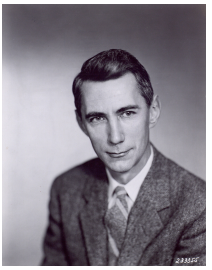

Figura 1.1: Claude Shannon

El problema de la teoría de códigos es encontrar una técnica óptima que nos permita corregir de forma segura los errores producidos en un mensaje enviado a través de un canal ruidoso. La idea es añadir información redundante al mensaje que nos facilite la corrección tras el envío. Esta información redundante se añade con un algoritmo y el mensaje que nos queda tras el cambio lo llamamos palabra del código.

[*] Claude Elwood Shannon 1916-2001, Estados Unidos. Se licenció en ingeniería eléctrica y matemáticas. Trabajó como investigador en el Instituto Tecnológico de Massachusetts (MIT). Es considerado el padre de la *Teoría de la información*.

En la Figura 1.2 está representado un ejemplo de comunicación utilizando códigos correctores. Nuestro objetivo es enviar un mensaje, representado en la figura por $\mathbf{m}$. Para ello, el mensaje se somete a un proceso de codificación donde se le añade la redundancia requerida y se genera una palabra del código $\mathbf{c}$. Tras esto, el mensaje, ya codificado, se envía a través de un canal que de ser ruidoso podrá generar un error $\mathbf{e}$. El mensaje recibido $\mathbf{y}$ será la palabra codificada más el error generado. Para concluir, pasamos al proceso de decodificación donde se eliminan los errores y las redundancias. El mensaje recibido $\hat{\mathbf{m}}$ es una estimación del mensaje original. Si hemos decodificado exitosamente, ambos coincidirán $\hat{\mathbf{m}} = \mathbf{m}$.

Para realizar todo este proceso es necesario prefijar un *alfabeto* que será un conjunto finito de letras y un *código*, denotado por $\mathscr{C}$, que será un conjunto finito de palabras. En general, nuestro alfabeto será el cuerpo finito de q elementos que denotaremos por $\mathbb{F}_q$. Por otro lado, el mensaje original será una k-upla de elementos del alfabeto escogido, es decir, un vector de $\mathbb{F}_q^k$. El proceso de codificación será una función de parámetros $[n, k]$, $\text{Enc}\colon \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$ con $k < n$ que transforma el mensaje $\mathbf{m} \in \mathbb{F}_q^k$ en una palabra del código $\mathbf{c} \in \mathscr{C} \subseteq \mathbb{F}_q^n$ añadiendo $n - k$ letras redundantes. La función de decodificación, por tanto, será $\text{Dec}\colon \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k$.

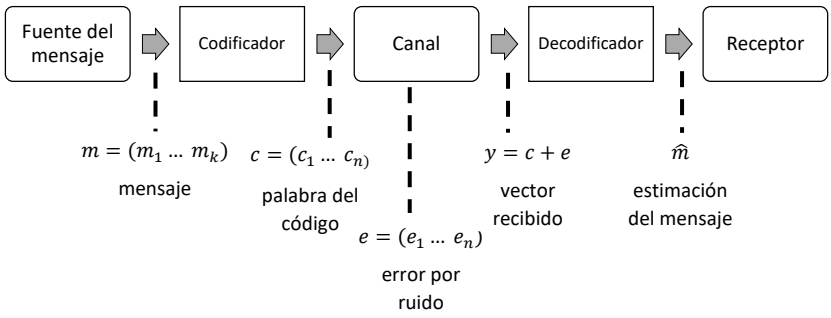

Figura 1.2: Ejemplo de comunicación utilizando códigos correctores.

1.2. SageMath

SageMath es un sistema de software libre de matemáticas. Su nombre procede de las siglas en inglés "Software for Algebra and Geometry Experimentation". Esta basado en el lenguaje Python y fue creado como una alternativa libre

de código abierto a Magma, Maple, Mathematica y Matlab. Podemos acceder a su página web utilizando el siguiente link. Este trabajo tiene como objetivo dar una visión general de lo que se puede hacer con SageMath en teoría de códigos y servir como pequeño tutorial de los principales métodos y clases específicas implementadas. En ningún caso, pretende ser un documento completo de todos los métodos y funcionalidades que ofrece SageMath en esta rama.

1.3. Códigos lineales

Denotaremos por $\mathbb{F}_q^n$ al espacio vectorial de las n-uplas definido sobre el cuerpo finito $\mathbb{F}_q$.

Definición 1.1 (Código lineal). *Un código lineal $\mathscr{C}$ sobre $\mathbb{F}_q$ es un subespacio de $\mathbb{F}_q^n$ de dimensión k. Para simplificar denotaremos este tipo de códigos como $[n,k]_q$ códigos.*

A los vectores $\mathbf{c} = (c_0, c_1, ..., c_{n-1})$ en $\mathscr{C}$ los llamaremos palabras. Como $\mathscr{C}$ es subespacio vectorial el vector cero es siempre una palabra del código. Además, es sencillo comprobar que el número de palabras de $\mathscr{C}$ es $M = q^k$. Denominaremos tasa de información a $\frac{k}{n}$, de esta forma, la redundancia añadida es $n - k$.

Todo subespacio se puede representar explícitamente a partir de una base o bien, utilizando sus ecuaciones implícitas. Por tanto, esto ocurrirá también con los códigos lineales. Por ello, las dos formas más comunes de presentar un código lineal $\mathscr{C}$ son a partir de la matriz generatriz y la matriz de paridad.

Definición 1.2 (Matriz generatriz). *Una matriz generatriz $\mathcal{G}$ para un $[n,k]_q$ código $\mathscr{C}$ es una matriz de tamaño $k \times n$ cuyas filas forman una base de $\mathscr{C}$.*

La matriz $\mathcal{G}$ no es única. Como $\mathcal{G}$ tiene rango k, es decir, tiene k columnas linealmente independientes, mediante operaciones por filas podemos reescribirla de forma que estas k columnas formen la matriz identidad I_k. Si además son exactamente las k primeras columnas, entonces decimos que la matriz generatriz $\mathcal{G}$ está escrita en forma estándar, es decir, $\mathcal{G} = (\mathrm{I}_k, \mathrm{A})$.

Definición 1.3. *Un vector de paridad para un $[n,k]_q$ código $\mathscr{C}$ es un vector $h \in \mathbb{F}_q^n$ que satisface la condición $\mathcal{G}h^T = 0$.*

Los vectores de paridad forman un subespacio de $\mathbb{F}_q^n$ de dimensión $n - k$. En efecto, el número de vectores de paridad linealmente independientes coincide con la dimensión de el núcleo de la aplicación $\varphi\colon \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$ definida como $\varphi(h) = \mathcal{G}h^T$. Por tanto, $\dim \ker(\varphi) = n - \dim \mathrm{Im}\,(\varphi) = n - \dim \mathscr{C} = n - k$.

Definición 1.4 (Matriz de paridad). *La matriz de paridad $\mathcal{H}$ para un $[n,k]_q$ código $\mathscr{C}$ es una matriz de tamaño $(n-k)\times n$ cuyas filas son vectores de paridad linealmente independientes.*

Observación 1.5. Sean $\mathcal{G}$ y $\mathcal{H}$ matrices generatriz y de paridad, respectivamente, de un código $\mathscr{C}$. Entonces $\mathcal{G}\mathcal{H}^T = 0$.

Gracias a la matriz de paridad podemos dar otra definición para $\mathscr{C}$,

$$\mathscr{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathcal{H}\mathbf{c}^T = 0\}.$$

Proposición 1.6. *Si $\mathcal{G} = (\mathrm{I}_k, \mathrm{A})$ es una matriz generatriz en forma estándar de un $[n,k]_q$ código $\mathscr{C}$. Entonces podemos escribir la matriz de paridad $\mathcal{H}$ de $\mathscr{C}$ como $\mathcal{H} = (-\mathrm{A}^T, \mathrm{I}_{n-k})$.*

Demostración. Es claro que $\mathcal{G}\mathcal{H}^T = A - A = 0$. Por otro lado, sea φ la aplicación $\varphi\colon \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^{n-k}$ definida como $\varphi(x) = \mathcal{H}x^T$ Entonces, $\mathscr{C} \subseteq \ker(\varphi)$. Además, $\dim\ker(\varphi) = n - \dim \mathrm{Im}\,(\varphi) = n - (n-k) = k = \dim \mathscr{C}$. □

La matriz de paridad $\mathcal{H}$ genera un código particular que llamaremos *código dual* de $\mathscr{C}$ y denotaremos por $\mathscr{C}^\perp$.

Veamos un ejemplo de cómo generar un código lineal a partir de las matrices generatriz y de paridad en SageMath.

Ejemplo 1.7. Consideramos las matrices,

$$\mathcal{G} = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,1 \\ 0\,1\,1\,1\,1\,1\,0 \end{pmatrix} \in \mathbb{F}_2^{2\times 7} \quad \text{y} \quad \mathcal{H} = \begin{pmatrix} 1\,1\,1\,0\,0\,0\,0 \\ 1\,1\,0\,1\,0\,0\,0 \\ 1\,1\,0\,0\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,0 \\ 1\,0\,0\,0\,0\,0\,1 \end{pmatrix} \in \mathbb{F}_2^{5\times 7}$$

Utilizaremos la matriz $\mathcal{G} \in \mathbb{F}_2^{2\times 7}$ como una matriz generatriz de $\mathscr{C}$, es decir $\mathcal{C}$ es un $[7,2]_2$ código.

```
Sage: G=matrix(GF(2),[[1,0,1,1,1,0,1], [0,1,1,1,1,1,0]])
Sage: C=LinearCode(G); C
>[7,2] linear code over GF(2)
```

También podemos utilizar la matriz de paridad $\mathcal{H}$ para generar el código $\mathcal{C}$.

```
Sage: H=matrix(GF(2),[[1,1,1,0,0,0,0], [1,1,0,1,0,0,0], [1,1,0,0,1,0,0],
                      [0,1,0,0,0,1,0], [1,0,0,0,0,0,1]])
Sage: C=codes.LinearCodeFromCheckMatrix(H); C
>[7,2] linear code over GF(2)
```

Comprobemos que las matrices $\mathcal{G}$ y $\mathcal{H}$ son matrices generatriz y de paridad, respectivamente, para $\mathscr{C}$ ya que cumplen que $\mathcal{G}\mathcal{H}^T = 0$.

```
Sage: P=G*H.transpose(); P
>[0 0 0 0 0]
 [0 0 0 0 0]
```

Ejemplo 1.8. En este ejemplo generaremos un $[8, 2]_3$ código $\mathscr{C}$ arbitrario, calculamos una matriz generatriz y una de paridad de $\mathscr{C}$ y comprobamos su longitud y dimensión.

```
Sage: C=codes.RandomLinearCode(8,2,GF(3)); C
>[8,2] linear code over GF(3)
Sage: G=C.generator_matrix(); G
>[2 2 0 0 1 2 0 0]
 [2 0 1 2 2 2 2 0]
Sage:H=C.parity_check_matrix(); H
>[1 0 0 0 0 2 0 0]
 [0 1 0 0 0 2 1 0]
 [0 0 1 0 0 0 1 0]
 [0 0 0 1 0 0 2 0]
 [0 0 0 0 1 1 1 0]
 [0 0 0 0 0 0 0 1]
Sage: C.length(), C.dimension()
>8,2
```

La siguiente función crea una matriz generatriz para $\mathscr{C}$ en forma sistemática.

```
Sage: G1=C.systematic_generator_matrix(); G1
>[1 0 2 1 1 1 1 0]
 [0 1 1 2 1 0 2 0]
```

Definición 1.9 (Código dual). *El código dual de un $[n, k]_q$ código $\mathscr{C}$ es*

$$\mathscr{C}^\perp = \{h \in \mathbb{F}^n \mid \langle h, \mathbf{c}\rangle = 0, \forall \mathbf{c} \in \mathscr{C}\}$$

donde $\langle x, y\rangle = x_0y_0 + x_1y_1 + \ldots + x_{n-1}y_{n-1}$ para $x, y \in \mathbb{F}_q^n$ denota el producto interior en $\mathbb{F}_q^n$.

Proposición 1.10. *Sea $\mathscr{C}$ un $[n, k]_q$ código. Entonces:*

i. Si $\mathcal{G}$ es una matriz generatriz para $\mathscr{C}$ entonces $\mathcal{G}$ es una matriz de paridad para el código dual $\mathscr{C}^\perp$.
ii. El código dual $\mathscr{C}^\perp$ es un $[n, n-k]_q$ código lineal sobre $\mathbb{F}$.
iii. $(\mathscr{C}^\perp)^\perp = \mathscr{C}$.

Demostración. i. Por definición de código dual:

$$h \in \mathscr{C}^{\perp} \iff \langle \mathbf{c}, h\rangle = 0,\ \forall \mathbf{c} \in \mathscr{C} \iff \mathbf{m}\mathcal{G}h^T = 0,\ \forall \mathbf{m} \in \mathbb{F}_q^k \iff Gh^T = 0.$$

Esto significa que $\mathscr{C}^{\perp}$ es el espacio nulo de $\mathcal{G}$. Como $\mathcal{G}$ es una matriz $k \times n$ de rango k y el espacio lineal $\mathscr{C}^{\perp}$ tiene dimension $n-k$, es fácil comprobar que $\mathcal{G}$ es una matriz de paridad para $\mathscr{C}^{\perp}$.

ii. Como $\mathcal{G}$ es una matriz de paridad de $\mathscr{C}^{\perp}$. Entonces, $\dim(\mathscr{C}^{\perp}) = n-k$.

iii. Sea $\mathbf{c} \in \mathscr{C}$. Entonces $\langle \mathbf{c}, h\rangle = 0$, $\forall h \in \mathscr{C}^{\perp}$. Por tanto, $\mathscr{C} \subseteq (\mathscr{C}^{\perp})^{\perp}$. Además, aplicando el apartado *i.* dos veces observamos que $\mathscr{C}$ y $(\mathscr{C}^{\perp})^{\perp}$ tienen la misma dimensión. De donde deducimos la igual de conjuntos.

□

Por la Proposición 1.10, diremos que h está en $\mathscr{C}^{\perp}$ si, y sólo si, $\mathcal{G}h^T = 0$.

Un concepto importante para un código $\mathscr{C}$ es la distancia mínima entre las palabras del código. Para ello se define el concepto de *distancia Hamming.*

Definición 1.11 (Distancia Hamming). *La distancia Hamming entre dos vectores $x, y \in \mathbb{F}_q^n$, denotada por $\mathrm{d}_H(x,y)$, es el número de coordenadas en las que x e y difieren.*

Se comprueba que $\mathrm{d}_H(x,y)$ cumple las propiedades de distancia,

- $\mathrm{d}_H(x,y) \geq 0$ para cada $x, y \in \mathbb{F}_q^n$
- $\mathrm{d}_H(x,y) = 0 \Leftrightarrow x = y$
- $\mathrm{d}_H(x,y) = \mathrm{d}_H(y,x)$ para cada $x, y \in \mathbb{F}_q^n$
- $\mathrm{d}_H(x,y) \leq \mathrm{d}_H(x,z) + \mathrm{d}_H(z,y)$ para cada $x, y, z \in \mathbb{F}_q^n$

Definición 1.12 (Distancia mínima). *La distancia mínima de un código $\mathscr{C}$, $\mathrm{d}_{min}(\mathscr{C})$ es la menor distancia Hamming entre cada par de palabras diferentes de un código, es decir,*

$$\mathrm{d}_{min}(\mathscr{C}) = min\{\mathrm{d}_H(\mathbf{c}_1, \mathbf{c}_2) \mid \mathbf{c}_1, \mathbf{c}_2 \in \mathscr{C},\ \mathbf{c}_1 \neq \mathbf{c}_2\}.$$

Otro concepto importante para un código $\mathscr{C}$ es el peso de cada una de sus palabras, lo llamaremos *peso Hamming* o simplemente *peso* y lo denotaremos por $\mathrm{w}_H(x)$.

Definición 1.13. *El soporte de un vector $x \in \mathbb{F}_q^n$, denotado por $\mathrm{supp}(x)$, es el conjunto de sus coordenadas distintas de cero, es decir, $\mathrm{supp}(x) = \{i \mid x_i \neq 0\}$.*

Definición 1.14 (Peso Hamming). *El peso Hamming de un vector $x \in \mathbb{F}_q^n$ es el número de coordenadas distintas de cero, es decir,*

$$\mathrm{w}_H(x) = \sharp\{i \mid x_i \neq 0\} = \sharp\mathrm{supp}(x),$$

donde $\sharp A$ denota el cardinal de A.

Definición 1.15 (Peso mínimo). *El peso mínimo de un código* $\mathscr{C}$, $\mathrm{w}_{min}(\mathscr{C})$ *es el menor peso distinto de cero entre todas las palabras del código, es decir,*

$$\mathrm{w}_{min}(\mathscr{C}) = min\{\mathrm{w}_H(\mathbf{c}) \mid \mathbf{c} \in \mathscr{C} \setminus \{0\}\}.$$

Lema 1.16. *Sea* $\mathscr{C}$ *un* $[n,k]_q$ *código entonces* $\mathrm{d}_{min}(\mathscr{C}) = \mathrm{w}_{min}(\mathscr{C})$.

Demostración. En primer lugar, observamos que $\mathrm{w}_H(x) = \mathrm{d}_H(x,0), \forall x \in \mathbb{F}_q^n$. Además, sabemos que $\mathrm{d}_H(x,y) = \mathrm{d}_H(x-z,y-z), \forall x,y,z \in \mathbb{F}_q^n$. En particular, $\mathrm{d}_H(x,y) = \mathrm{d}_H(x-y,y-y) = \mathrm{d}_H(x-y,0) = \mathrm{w}_H(x-y)$. □

Veamos cómo calcular en SageMath el código dual y la distancia mínima de $\mathscr{C}$.

Ejemplo 1.17. Continuamos con el ejemplo 1.7. Calculamos $\mathscr{C}^\perp$ y generamos una de sus matrices generatrices, que será matriz de paridad para $\mathscr{C}$.

```
Sage: C2=C.dual_code()
>[7, 5] linear code over GF(2)
Sage: H2=C2.generator_matrix()
>[1 0 0 0 0 0 1]
 [0 1 0 0 0 1 0]
 [0 0 1 0 0 1 1]
 [0 0 0 1 0 1 1]
 [0 0 0 0 1 1 1]
Sage: P=G*H2.transpose(); P
>[0 0 0 0 0]
 [0 0 0 0 0]
```

Con la siguiente línea de comando, además de su distancia mínima, obtendremos el tiempo que tarda el CPU en calcular la distancia mínima de $\mathscr{C}$, recordemos que este algoritmo consiste en comparar todos los posibles pares de palabras del código.

```
Sage: %time C.minimum_distance()
>CPU times: user 31 ms, sys: 14.8 ms, total: 45.7 ms
 Wall time: 75.3 ms
>4
```

Lema 1.18. *Sean* $\mathscr{C}$ *un* $[n,k]_q$ *código lineal con matriz de paridad* $\mathcal{H}$. *Entonces* $d = \mathrm{d}_{min}(\mathscr{C})$ *es el menor entero tal que* d *columnas de* $\mathcal{H}$ *son linealmente dependientes.*

Demostración. Sean $h_1, \ldots, h_n$ las columnas de $\mathcal{H}$. Consideremos $\mathbf{c} \in \mathscr{C} \setminus \{0\}$ con $\mathrm{w}_H(\mathbf{c}) = \mathrm{w}$ y $\mathrm{supp}(c) = \{j_1, \ldots, j_\mathrm{w}\}$ donde $1 \leq j_1 < \cdots < j_\mathrm{w} \leq n$. Sabemos que $\mathcal{H}\mathbf{c}^T = 0$, por lo tanto,

$$c_{j_1}h_{j_1} + \ldots + c_{j_{\mathrm{w}}}h_{j_{\mathrm{w}}} = 0 \text{ con } c_{j_i} \neq 0, \forall i = 1, \ldots, \mathrm{w}.$$

Es decir, las columnas $h_{j_1}, \ldots, h_{j_{\mathrm{w}}}$ son linealmente dependientes.

Por otro lado, si $h_{j_1}, \ldots, h_{j_{\mathrm{w}}}$ son linealmente dependientes, entonces existen constantes $a_1, \ldots, a_{\mathrm{w}}$ no todas cero tales que $a_1h_{j_1} + \ldots + a_{\mathrm{w}}h_{j_{\mathrm{w}}} = 0$. Sea $\mathbf{c} \in \mathscr{C}$ tal que $c_j = 0, \forall j \neq j_i$ y $c_j = a_i$ si $j = j_i$ con $1 \leq i \leq w$. Es fácil comprobar que $\mathcal{H}\mathbf{c}^T = 0$, es decir, $\mathbf{c} \in \mathscr{C} \setminus \{0\}$ con $\mathrm{w}_H(\mathbf{c}) \geq \mathrm{w}$. □

Teorema 1.19 (Cota de Singleton). *Sea $\mathscr{C}$ un $[n,k]_q$ código lineal sobre el cuerpo finito $\mathbb{F}_q$ entonces $\mathrm{d}_{min}(\mathscr{C}) \leq n - k + 1$.*

Demostración. Cada $(n-k) \times (n-k+1)$ submatriz de una matriz de paridad $\mathcal{H}$ tiene rango menor o igual que $n-k$, entonces cada conjunto de $n-k+1$ columnas de la matriz de paridad son linealmente dependientes. Utilizando el Lema 1.18 concluimos el resultado. □

Definición 1.20 (Códigos MDS). *Los códigos que igualan la cota de Singleton se definen como códigos separables de máxima distancia, las siglas **MDS** provienen de su traducción en inglés "maximum distance separable".*

Proposición 1.21. *[11, Theorem 2.4.3]. Sea $\mathscr{C}$ un $[n,k]_q$ código MDS entonces $\mathscr{C}^\perp$ también es un código MDS de parámetros $[n, n-k]$ sobre $\mathbb{F}_q$.*

En el Capítulo 2 presentaremos una familia de códigos llamados Reed-Solomon generalizados que son MDS (Definición 2.61).

1.4. Codificación

Tomamos $\mathscr{C}$ un $[n,k]_q$ código lineal con matriz generatriz $\mathcal{G}$. Para codificar un mensaje $\mathbf{m} \in \mathbb{F}_q^k$ utilizaremos la función de codificación definida por:

$$\begin{aligned} \mathrm{Enc}\colon \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ \mathbf{m} &\longmapsto \mathbf{m}\mathcal{G} \in \mathscr{C}. \end{aligned}$$

Es decir, nuestra palabra codificada $\mathbf{c} \in \mathscr{C}$ será $\mathbf{c} = \mathbf{m}\mathcal{G}$.

Si la matriz generatriz $\mathcal{G}$ está en forma estándar $\mathcal{G} = (\mathrm{I}_k, \mathrm{A})$ entonces las primeras k entradas de la palabra transmitida $\mathbf{c}$ contendrán exactamente el mensaje $\mathbf{x}$. Los siguientes $n-k$ símbolos son la redundancia añadida a $\mathbf{m}$ que tienen el objetivo de ayudar a recuperar el mensaje $\mathbf{m}$ si ocurriesen errores en la transmisión.

Ejemplo 1.22. Continuamos con el Ejemplo 1.7. Consideremos $v = (0,1) \in \mathbb{F}_q^k$, codificaremos v multiplicándolo por una matriz generatriz del código $\mathscr{C}$.

$$v\mathcal{G} = (0,1)\begin{pmatrix} 1\,0\,1\,1\,1\,0\,1 \\ 0\,1\,1\,1\,1\,1\,0 \end{pmatrix} = (0,1,1,1,1,1,0)$$

Es decir,

```
Sage: word=vector(GF(2),[0,1])
Sage: codeword=word*G; codeword
>(0, 1, 1, 1, 1, 1, 0)
```

Otra forma de codificar en SageMath es utilizando el comando ENCODER. Se puede comprobar que ambos métodos dan el mismo resultado.

```
Sage: word=vector(GF(2),[0,1])
Sage: E = C.encoder()
Sage: codeword=E.encode(word); codeword
>(0, 1, 1, 1, 1, 1, 0)
```

1.5. Problema de decodificación

Sea $\mathscr{C}$ un $[n,k]_q$ código lineal con matriz generatriz $\mathcal{G}$. Sea $\mathbf{y} = \mathbf{c} + \mathbf{e}$ un vector recibido, con $\mathbf{c} \in \mathscr{C}$ y $\mathbf{e}$ el error generado. La función de decodificación se define como:

$$\begin{aligned} \mathrm{Dec}\colon \quad \mathbb{F}_q^n \quad &\longrightarrow \mathbb{F}_q^k \\ \mathbf{y} = \mathbf{c} + \mathbf{e} &\longmapsto \mathrm{Dec}(\mathbf{y}) = \mathbf{m},\ \mathbf{m}\mathcal{G} \in \mathscr{C}. \end{aligned}$$

El proceso de decodificación consiste en determinar qué palabra del código $\mathbf{c}$ y, por tanto, qué mensaje $\mathbf{m}$, es la que tiene más probabilidad de haber sido enviada si nosotros hemos recibido el vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$. Esta decodificación se conoce como decodificación por probabilidad maximal con abreviación **MLD**, por su traducción del inglés "maximum likelihood decoding". Lo que se busca, por tanto, es minimizar:

$$P_r\left(\frac{\mathbf{y} \text{ recibida}}{\mathbf{c} \text{ enviada}}\right).$$

Por otro lado, existe la decodificación por mínimas distancias, con abreviación **MDD** por su traducción al inglés "minimum distance decoding", que consiste en, dado un vector recibido $\mathbf{y}$, encontrar una palabra $\mathbf{c}$ que minimice $\mathrm{d}_H(\mathbf{y}, \mathbf{c})$. Se demuestra que en canales simétricos* **MDD** es equivalente a **MLD**.

* Sean $\mathcal{X}$ un alfabeto de entrada e $\mathcal{Y}$ un alfabeto de salida para un canal. Definimos:

1. Canal discreto. Si $\mathcal{X}$ e $\mathcal{Y}$ son finitos.
2. Canal sin memoria. Cuando la salida en el instante i depende únicamente de la entrada en el instante i.
3. Canales simétricos. Cuando la probabilidad de transmisión está dada por

$$P_r\left(\frac{y \in \mathcal{Y}}{x \in \mathcal{X}}\right) = \begin{cases} p & \text{,si } x = y \\ \frac{1-p}{|y|-1} & \text{,si } x \neq y \end{cases}$$

Proposición 1.23. *En canales simétricos* **MDD** = **MLD**.

Demostración. Probaremos un caso particular. En canales simétricos binarios con probabilidad de error $p < \frac{1}{2}$, **MDD**=**MLD**. Sea $\mathrm{d}_H(x, y) = d$. Entonces,

$$P_r\left(\frac{\mathbf{y} \text{ recibida}}{\mathbf{c} \text{ enviada}}\right) = (1-p)^{n-d}p^d = (1-p)^n\left(\frac{p}{1-p}\right)^d.$$

Como $p < \frac{1}{2}$ la probabilidad se maximiza cuando d es mínimo. □

La búsqueda de algoritmos de decodificación eficientes es una de las temáticas más activas en la investigación de la teoría de códigos por sus aplicaciones prácticas. En general, codificar es un proceso sencillo, sin embargo, decodificar es un proceso complejo si el código tiene un tamaño razonablemente largo. El proceso de decodificación es el más delicado en una comunicación con códigos correctores.

Definición 1.24 (Complejidad). *En computación, sea $\mathcal{A}$ un algoritmo que tiene como entrada una palabra binaria. Entonces el tiempo o complejidad de trabajo $\mathcal{O}_{\mathrm{T}}(\mathcal{A}; n)$ es el número de operaciones elementales necesarias en función de la longitud n que se realizan en el algoritmo $\mathcal{A}$ para obtener el resultado. El espacio o complejidad de memoria $\mathcal{O}_{\mathrm{S}}(\mathcal{A}; n)$ es el número máximo de bits de memoria necesarios durante la ejecución del algoritmo con una entrada de n bits. La complejidad $\mathcal{O}(\mathcal{A}; n)$ es el máximo de $\mathcal{O}_{\mathrm{T}}(\mathcal{A}; n)$ y $\mathcal{O}_{\mathrm{S}}(\mathcal{A}; n)$.*

La primera idea de algoritmo de decodificación que se nos presenta es la *decodificación por fuerza bruta.*

Decodificación por fuerza bruta.

Sea $\mathbf{y}$ una palabra recibida. Entonces, el método de decodificación por fuerza bruta consiste en calcular la distancia de Hamming $\mathrm{d}_H(\mathbf{y}, \mathbf{c})$, para todas las palabras del código $\mathbf{c} \in \mathscr{C}$ y nos devuelve la que minimice dicha distancia de Hamming. Su complejidad es $\mathcal{O}(nq^k)$ pues hay q^k palabras de longitud n.

1.5.1. Decodificación por síndrome

Otro algoritmo de decodificación eficiente pero costoso para códigos lineales es el *algoritmo de decodificación por síndrome.* Este algoritmo es uno de los algoritmos más utilizados en la literatura.

Definición 1.25 (Síndrome). *Sea $\mathcal{H}$ una matriz de paridad para el $[n, k]_q$ código $\mathscr{C}$. El síndrome de $x \in \mathbb{F}_q^n$ es $\mathcal{S}(x) = \mathcal{H}x^T$.*

Observación 1.26. El síndrome de una palabra $\mathbf{c} \in \mathscr{C}$ es cero por definición.

Lema 1.27. *Sean $\mathscr{C}$ un $[n,k]_q$ código, $\mathbf{y} = \mathbf{c}+\mathbf{e}$ una palabra recibida, con $\mathbf{c} \in \mathscr{C}$ y $\mathbf{e}$ el error generado. Entonces, $\mathcal{S}(\mathbf{y}) = \mathcal{S}(\mathbf{e})$.*

Demostración. Teniendo en cuenta la Observación 1.26 se tiene que:

$$\mathcal{S}(\mathbf{y}) = \mathcal{S}(\mathbf{c}+\mathbf{e}) = \mathcal{S}(\mathbf{c}) + \mathcal{S}(\mathbf{e}) = \mathcal{S}(\mathbf{e}).$$

□

Sea $\mathcal{H}$ una matriz de paridad de un $[n,k]_q$ código $\mathscr{C}$, teniendo en cuenta la definición de síndrome podemos inducir una clase de equivalencia en $\mathscr{C}$:

$$\forall x, y \in \mathbb{F}_q^n,\ x \sim y \iff \exists c \in \mathscr{C} : x = y + c.$$

Es decir, $x, y \in a + \mathscr{C} \iff \mathcal{H}(x^T) = \mathcal{H}(y^T)$, donde $a + \mathscr{C}$ representa la clase de equivalencia de a. A partir de esta clase de equivalencia, podemos particionar $\mathbb{F}_q^n$ en q^{n-k} clases de equivalencia distintas, cada una con q^k elementos.

Teorema 1.28 (Teorema de Lagrange). *Sea $\mathscr{C}$ un $[n,k]_q$ código. Entonces*

a. Cada clase de equivalencia contiene exactamente q^k elementos.
b. Dos clases de equivalencia o son iguales o son disjuntas.

Demostración. *a.* Sea ϕ la siguiente aplicación biyectiva: $\begin{array}{rcl} \phi\colon \mathbb{F}_q^n[x] & \longrightarrow & \mathbb{F}_q^n \\ x & \longmapsto & x + a \end{array}$
donde $a \notin \mathscr{C}$. Entonces, $\sharp(a + \mathscr{C}) = \sharp\mathscr{C} = q^k$.

b. Supongamos que $v \in (a + \mathscr{C}) \cap (b + \mathscr{C})$ es decir, $v = a + c_1 = b + c_2$, con $c_1, c_2 \in \mathscr{C}$. Entonces,

$$\left.\begin{array}{l} b = a + (c_1 - c_2) \in a + \mathscr{C} \Longrightarrow b + \mathscr{C} \subseteq a + \mathscr{C} \\ a = b + (c_1 - c_2) \in b + \mathscr{C} \Longrightarrow a + \mathscr{C} \subseteq b + \mathscr{C} \end{array}\right\} \quad \Longrightarrow a + \mathscr{C} = b + \mathscr{C}.$$

□

Definición 1.29 (Elemento líder). *Llamamos elementos líderes a las palabras que tienen menor peso Hamming en una clase de equivalencia. Y a dicho peso lo llamaremos peso de la clase de equivalencia.*

El vector cero es el único líder de la clase $0 + \mathscr{C} = \mathscr{C}$. El elemento líder no tiene porqué ser único.

Proposición 1.30. *Toda clase de equivalencia con peso $\mathrm{w}_H \leq t = \left\lfloor \frac{d-1}{2} \right\rfloor$ tiene un único elemento líder siendo $d = \mathrm{d}_{min}(\mathscr{C})$.*

Demostración. Procedamos por reducción al absurdo.
Supongamos que $\exists a, b \in x + \mathscr{C}$: $\mathrm{w}_H(a) \leq t, \mathrm{w}_H(b) \leq t$. Entonces,

$$\begin{cases} a \in x + \mathscr{C} \Longrightarrow a = x + c_1, & \text{con } c_1 \in \mathscr{C} \\ b \in x + \mathscr{C} \Longrightarrow b = x + c_2, & \text{con } c_2 \in \mathscr{C} \end{cases}$$

Por tanto,

$$\mathrm{d}_H(c_1, c_2) = \mathrm{d}_H(a-x, b-x) = \mathrm{w}_H(a-b) \leq \mathrm{w}_H(a)+\mathrm{w}_H(b) \leq 2t = 2(\tfrac{d-1}{2}) = d-1.$$

□

Decodificación por síndrome. Supongamos que $\mathbf{y} = \mathbf{c}+\mathbf{e}$ con $\mathbf{c} \in \mathscr{C}$ es la palabra recibida. Para realizar la decodificación por síndrome de $\mathbf{y}$ debemos ejecutar los siguientes pasos:

1. Construimos una tabla que llamaremos tabla de síndromes y que se compondrá de los siguientes elementos:
   - Síndromes. En la primera columna aparecen todos los posibles síndromes del código.
   - Elemento líder. Escogemos un elemento líder para cada clase de equivalencia y lo colocamos en la segunda columna en su fila correspondiente.
2. Calculamos el síndrome de la palabra recibida $\mathcal{S}(\mathbf{y})$.
3. Buscamos en la tabla el elemento líder $\mathbf{e} \in \mathbb{F}_q^n$: $\mathcal{S}(\mathbf{y}) = \mathcal{S}(\mathbf{e})$.
4. Decodificamos $\mathbf{y} \in \mathbb{F}_q^n$ utilizando que $\mathbf{y} - \mathbf{e} \in \mathscr{C}$.

La complejidad del método de decodificación por síndrome es $\mathcal{O}((n-k)q^{n-k})$ pues, en este caso, tenemos q^{n-k} síndromes distintos de longitud $(n-k)$.

Veamos un ejemplo de decodificación por síndrome utilizando SageMath.

Ejemplo 1.31. Sea $\mathscr{C}$ el código del ejemplo 1.7. Supongamos que recibimos la palabra $\mathbf{y} = \mathbf{c} + \mathbf{e} = (1, 1, 1, 0, 1, 1, 0)$.

1. Calculamos la tabla de síndrome (Figura 1.3). Observamos que no es posible garantizar que el elemento líder sea único cuando el peso de éste sea mayor estricto que $t = \left\lfloor \frac{4-1}{2} \right\rfloor = 1$. SageMath nos permite generar la tabla de síndromes escogiendo un único líder (Figura 1.3).
2. Calculamos el síndrome de la palabra $\mathbf{y}$.

```
Sage: y=vector(GF(2),[1,1,1,0,1,1,0])
Sage: S=y*H.transpose(); S
>(1,0,1,0,1)
```

3. Buscamos en la tabla el elemento líder del síndrome generado, $\mathbf{e} = (0, 0, 0, 1, 0, 0, 1)$.
4. Decodificamos.

```
Sage: e=vector(GF(2),[0,0,0,1,0,0,1])
Sage: c=y-e; c
>(1,1,1,1,1,1,1)
```

Por tanto, nuestra palabra enviada es: $\mathbf{c} = (1, 1, 1, 1, 1, 1, 1)$.

Síndrome	Elementos líderes
(0, 0, 0, 0, 0)	(0, 0, 0, 0, 0, 0, 0)
(0, 0, 0, 0, 1)	(0, 0, 0, 0, 1, 0, 0)
(0, 0, 0, 1, 0)	(0, 0, 0, 1, 0, 0, 0)
(0, 0, 0, 1, 1)	(0, 0, 0, 1, 1, 0, 0)
(0, 0, 1, 0, 0)	(0, 0, 1, 0, 0, 0, 0)
(0, 0, 1, 0, 1)	(0, 0, 1, 0, 1, 0, 0)
(0, 0, 1, 1, 0)	(0, 0, 1, 1, 0, 0, 0)
(0, 0, 1, 1, 1)	(1, 0, 0, 0, 0, 0, 1), (1, 0, 0, 0, 0, 0, 1)
(0, 1, 0, 0, 0)	(0, 1, 0, 0, 0, 0, 0)
(0, 1, 0, 0, 1)	(0, 1, 0, 0, 1, 0, 0)
(0, 1, 0, 1, 0)	(0, 1, 0, 1, 0, 0, 0)
(0, 1, 0, 1, 1)	(0, 0, 1, 0, 0, 1, 0)
(0, 1, 1, 0, 0)	(0, 1, 1, 0, 0, 0, 0)
(0, 1, 1, 0, 1)	(0, 0, 0, 1, 0, 1, 0)
(0, 1, 1, 1, 0)	(0, 0, 0, 0, 1, 1, 0)
(0, 1, 1, 1, 1)	(0, 0, 0, 0, 0, 1, 0)
(1, 0, 0, 0, 0)	(1, 0, 0, 0, 0, 0, 0)
(1, 0, 0, 0, 1)	(1, 0, 0, 0, 1, 0, 0)
(1, 0, 0, 1, 0)	(1, 0, 0, 1, 0, 0, 0)
(1, 0, 0, 1, 1)	(0, 0, 1, 0, 0, 0, 1)
(1, 0, 1, 0, 0)	(1, 0, 1, 0, 0, 0, 0)
(1, 0, 1, 0, 1)	(0, 0, 0, 1, 0, 0, 1)
(1, 0, 1, 1, 0)	(0, 0, 0, 0, 1, 0, 1)
(1, 0, 1, 1, 1)	(0, 0, 0, 0, 0, 0, 1)
(1, 1, 0, 0, 0)	(1, 1, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 1, 1)
(1, 1, 0, 0, 1)	(1, 1, 0, 0, 1, 0, 0), (0, 0, 0, 0, 1, 1, 1)
(1, 1, 0, 1, 0)	(1, 1, 0, 1, 0, 0, 0), (0, 0, 0, 1, 0, 1, 1)
(1, 1, 0, 1, 1)	(1, 0, 1, 0, 0, 1, 0), (0, 1, 1, 0, 0, 0, 1)
(1, 1, 1, 0, 0)	(1, 1, 1, 0, 0, 0, 0), (0, 0, 1, 0, 0, 1, 1)
(1, 1, 1, 0, 1)	(1, 0, 0, 1, 0, 1, 0), (0, 1, 0, 1, 0, 0, 1)
(1, 1, 1, 1, 0)	(1, 0, 0, 0, 1, 1, 0), (0, 1, 0, 0, 1, 0, 1)
(1, 1, 1, 1, 1)	(1, 0, 0, 0, 0, 1, 0), (0, 1, 0, 0, 0, 0, 1)

```
Sage: D = codes.decoders.LinearCodeSyndromeDecoder(C)
      D.syndrome_table()
>{(0, 0, 0, 0, 0): (0, 0, 0, 0, 0, 0, 0),
  (0, 0, 0, 0, 1): (0, 0, 0, 0, 1, 0, 0),
  (0, 0, 0, 1, 0): (0, 0, 0, 1, 0, 0, 0),
  (0, 0, 0, 1, 1): (0, 0, 0, 1, 1, 0, 0),
  (0, 0, 1, 0, 0): (0, 0, 1, 0, 0, 0, 0),
  (0, 0, 1, 0, 1): (0, 0, 1, 0, 1, 0, 0),
  (0, 0, 1, 1, 0): (0, 0, 1, 1, 0, 0, 0),
  (0, 0, 1, 1, 1): (1, 0, 0, 0, 0, 0, 1),
  (0, 1, 0, 0, 0): (0, 1, 0, 0, 0, 0, 0),
  (0, 1, 0, 0, 1): (0, 1, 0, 0, 1, 0, 0),
  (0, 1, 0, 1, 0): (0, 1, 0, 1, 0, 0, 0),
  (0, 1, 0, 1, 1): (0, 0, 1, 0, 0, 1, 0),
  (0, 1, 1, 0, 0): (0, 1, 1, 0, 0, 0, 0),
  (0, 1, 1, 0, 1): (0, 0, 0, 1, 0, 1, 0),
  (0, 1, 1, 1, 0): (0, 0, 0, 0, 1, 1, 0),
  (0, 1, 1, 1, 1): (0, 0, 0, 0, 0, 1, 0),
  (1, 0, 0, 0, 0): (1, 0, 0, 0, 0, 0, 0),
  (1, 0, 0, 0, 1): (1, 0, 0, 0, 1, 0, 0),
  (1, 0, 0, 1, 0): (1, 0, 0, 1, 0, 0, 0),
  (1, 0, 0, 1, 1): (0, 0, 1, 0, 0, 0, 1),
  (1, 0, 1, 0, 0): (1, 0, 1, 0, 0, 0, 0),
  (1, 0, 1, 0, 1): (0, 0, 0, 1, 0, 0, 1),
  (1, 0, 1, 1, 0): (0, 0, 0, 0, 1, 0, 1),
  (1, 0, 1, 1, 1): (0, 0, 0, 0, 0, 0, 1),
  (1, 1, 0, 0, 0): (1, 1, 0, 0, 0, 0, 0),
  (1, 1, 0, 0, 1): (1, 1, 0, 0, 1, 0, 0),
  (1, 1, 0, 1, 0): (1, 1, 0, 1, 0, 0, 0),
  (1, 1, 0, 1, 1): (1, 0, 1, 0, 0, 1, 0),
  (1, 1, 1, 0, 0): (1, 1, 1, 0, 0, 0, 0),
  (1, 1, 1, 0, 1): (1, 0, 0, 1, 0, 1, 0),
  (1, 1, 1, 1, 0): (1, 0, 0, 0, 1, 1, 0),
  (1, 1, 1, 1, 1): (1, 0, 0, 0, 0, 1, 0)}
```

Figura 1.3: Tabla de síndrome realizada manualmente y código de Sagemath que crea dicha tabla.

Lema 1.32. *Sea $\mathscr{C}$ un $[n,k]_q$ código entonces, el número de palabras de peso menor o igual que $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, con $d = \mathrm{d}_{min}(\mathscr{C})$ donde $\lfloor x \rfloor$ denota al menor entero más próximo a x, es:*

$$1 + (q-1)\binom{n}{1} + (q-1)^2\binom{n}{2} + \ldots + (q-1)^t\binom{n}{t} = \sum_{j=0}^{t}(q-1)^j\binom{n}{j}$$

Demostración. El resultado es consecuencia de una demostración combinatoria. □

Teorema 1.33 (Cota de Hamming). *Sea $\mathscr{C}$ un $[n,k]_q$ código entonces el número de vectores de peso menor o igual que $t = \left\lfloor \frac{d-1}{2} \right\rfloor$, con $d = \mathrm{d}_{min}(\mathscr{C})$ se puede acotar por q^{n-k}, es decir,*

$$\sum_{j=0}^{t}(q-1)^j\binom{n}{j} \leq q^{n-k}$$

Los códigos que alcanzan esta cota se llaman códigos perfectos.

Demostración. Todos los vectores de peso menor o igual que t son el único líder de su clase de equivalencia por la Proposición 1.30. Por tanto, el número de vectores de peso menor o igual que t es más pequeño que el número total de clases de equivalencia q^{n-k}. □

Proposición 1.34. *Un $[n,k]_q$ código lineal $\mathscr{C}$ es capaz de corregir hasta t errores por **MDD** si, y sólo si, $\mathrm{t} \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, donde $d = \mathrm{d}_{min}$. En este caso, se dice que el código es t-corrector de errores.*

Demostración. Sea $\mathrm{t} \leq \left\lfloor \frac{d-1}{2} \right\rfloor$. Procedemos por reducción al absurdo, suponga-mos que recibimos $\mathbf{y} \in \mathbb{F}_q^n$ y que el algoritmo **MDD** no nos permite corregir t errores. Es decir, $\exists \mathbf{c}_1, \mathbf{c}_2 \in \mathscr{C}$ y $\mathbf{e}_1, \mathbf{e}_2 \in \mathbb{F}_q^n$: $\mathbf{y} = \mathbf{c}_1 + \mathbf{e}_1 = \mathbf{c}_2 + \mathbf{e}_2$ con peso $\mathrm{w}_H(\mathbf{e}_1) \leq t$ y $\mathrm{w}_H(\mathbf{e}_2) \leq t$. Entonces, $\mathrm{w}_H(\mathbf{e}_1 - \mathbf{e}_2) = \mathrm{w}_H(\mathbf{c}_1 - \mathbf{c}_2) \leq 2t \leq d-1$ contradiciendo que $\mathrm{d}_{min} = d$.

Supongamos ahora que $t > \left\lfloor \frac{d-1}{2} \right\rfloor$. Sea $\mathbf{y} \in \mathbb{F}_q^n$ tal que $\mathbf{y} = \mathbf{c} + \mathbf{e}$ con $\mathbf{c} \in \mathscr{C}$ y $\mathbf{e}$ el error generado. Consideramos el vector error definido por $e_i = -c_i$, $\forall i \in I$ tales que $c_i \neq 0$ y $|I| = t+1$. Entonces,

$$\mathrm{d}_H(0,\mathbf{y}) = \mathrm{w}_H(\mathbf{y}) = \mathrm{w}_H(\mathbf{c}) + \mathrm{w}_H(\mathbf{e}) = d - (t+1) = \tfrac{d-1}{2} \leq t$$

pero $\mathrm{d}_H(\mathbf{c},\mathbf{y}) = \mathrm{w}_H(\mathbf{e}) = t+1$. Al decodificar por **MDD**, la decodificación del vector $\mathbf{y}$ nos queda $\mathrm{dec}(\mathbf{y}) = 0 \neq \mathbf{c}$. □

Tipos de decodificadores

Podemos distinguir dos tipos de decodificadores:

1. **Decodificador único**. Busca la única palabra del código que minimice la distancia de Hamming con la palabra recibida. Si el número de errores supera la capacidad correctora del código entonces, o nos dice que hubo un fallo o nos devuelve una palabra errónea.
2. **Decodificador completo**. Siempre devuelve una de las palabras más cercanas al código. Un ejemplo de esta decodificación es la vista en el ejemplo 1.31 ya que aunque en algunos casos el elemento líder no sea único se devuelve un resultado.

2

Códigos cíclicos

Los *códigos cíclicos* han sido de gran interés en la teoría de códigos desde su comienzo ya que para longitudes relativamente pequeñas se consiguen códigos cíclicos con buenos parámetros. Es decir, fijadas la longitud y la dimensión, en algunos casos, los códigos con mejor distancia mínima son códigos cíclicos. Una pregunta aún abierta es si, cuando la longitud del código tiende a infinito, la familia de códigos cíclicos continua siendo buena.

Sea Ω un conjunto tal que $\sharp\Omega = n \in \mathbb{N}$, definimos el grupo de permutaciones $S_n = \{f\colon \Omega \longrightarrow \Omega \mid \text{f biyectiva}\}$. Sea $\sigma \in S_n$ decimos que es una permutación cíclica si para $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ se tiene que $\sigma(c) = (c_{n-1}, c_0, \ldots, c_{n-2})$

Definición 2.1 (Códigos cíclicos). *Un código lineal es cíclico si es invariante por permutaciones cíclicas, es decir, si para toda palabra del código $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathscr{C}$ se tiene que $\sigma(c) = (c_{n-1}, c_0, \ldots, c_{n-2}) \in \mathscr{C}$.*

Los subespacios $\{0\}$ y $\mathbb{F}_q^n$ son cíclicos y se les llama códigos cíclicos triviales.

2.1. Caracterización de los códigos cíclicos como ideales

Denotaremos por $\mathbb{F}_q[x]$ al anillo de polinomios con coeficientes en el cuerpo finito $\mathbb{F}_q$ y por $\mathbb{F}_q[x]_{<n}$ al espacio de polinomios de grado menor que n con coeficientes en $\mathbb{F}_q$. Además, $\deg(p(x))$ denotará el grado del polinomio $p(x)$.

El siguiente resultado nos permite trabajar con códigos cíclicos como ideales del anillo $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Para ello, es conveniente utilizar la enumeración de las coordenadas de códigos cíclicos comenzando en 0 y terminando en $n-1$ ya que los *códigos cíclicos* se escribirán como polinomios de grado menor que n. Es decir, podremos utilizar notación vectorial o polinómica de forma indiferente.

Lema 2.2. *La aplicación*

$$\begin{array}{rcl}\varphi\colon \mathbb{F}_q[x] & \longrightarrow & \mathbb{F}_q[x]_{<n} \\ f & \longmapsto & R(f, x^n-1)\end{array}$$

es un epimorfismo de $\mathbb{F}_q$-álgebras con $\ker\varphi = \langle x^n-1\rangle$*, donde $\langle a\rangle$ representa al ideal generado por a y $R(f,g)$ denota el resto de la división de f frente a g. Además, los elementos de $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ verifican:*

$$f_1 \equiv f_2 \mod (x^n-1) \iff R(f_1, x^n-1) = R(f_2, x^n-1).$$

Demostración. Es claro que φ está bien definida. Procedemos a ver que φ es un homomorfismo de espacios vectoriales. Por la división euclídea, $\forall f, g \in \mathbb{F}_q[x]$:

$$\begin{cases} f = p_1(x)(x^n-1) + R(f, x^n-1) \\ g = p_2(x)(x^n-1) + R(g, x^n-1) \\ f+g = p(x)(x^n-1) + R(f+g, x^n-1). \end{cases}$$

Entonces,
$p(x)(x^n-1)+R(f+g,x^n-1) = (p_1(x)+p_2(x))(x^n-1)+R(f,x^n-1)+R(g,x^n-1)$
$\iff (p(x)-p_1(x)-p_2(x))(x^n-1) = -R(f+g,x^n-1)+R(f,x^n-1)+R(g,x^n-1).$

Por tanto, como la parte derecha de la igualdad es un polinomio de grado mayor que n y la parte izquierda es un polinomio de grado menor que n. Para que se cumpla la igualdad es necesario que ambos sean el polinomio cero. Es decir, $R(f, x^n-1) + R(g, x^n-1) = R(f+g, x^n-1)$. De donde deducimos que, $\varphi(f) + \varphi(g) = \varphi(f+g)$. De igual forma se comprueba que $\varphi(\lambda f) = \lambda\varphi(f)$.

Comprobaremos ahora que φ es una aplicación sobreyectiva. Esto es, sea $f \in \mathbb{F}_q[x]_{<n}$, $\deg f = n$, por tanto, $R(f, x^n-1) = f$, es decir, $\varphi(f) = f$.

Para terminar, calculamos el núcleo de φ.

$$\begin{aligned}\ker\varphi &= \{f \in \mathbb{F}_q[x]_{<n} \mid \varphi(f) = 0\} = \{f \in \mathbb{F}_q[x]_{<n} \mid R(f, x^n-1) = 0\} = \\ &= \{f \in \mathbb{F}_q[x] \mid f(x) = p(x)(x^n-1), \text{ con } p(x) \in \mathbb{F}_q[x]\} = \langle x^n-1\rangle .\end{aligned}$$

Además, por el primer teorema de isomorfía,

$$\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle} \simeq \mathbb{F}_q[x]_{<n}.$$

□

La aplicación ψ definida como:

$$\begin{array}{rccccc}\psi\colon & \mathbb{F}_q^n & \longrightarrow & \mathbb{F}_q[x]_{<n} & \longrightarrow & \frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle} \\ & (a_0,\ldots,a_{n-1}) & \longmapsto & a_0+a_1x+\ldots+a_{n-1}x^{n-1} & \longmapsto & R(f(x), x^n-1)\end{array} \tag{2.1}$$

es, por tanto, un isomorfismo entre espacios vectoriales.

Con los siguientes resultados concluiremos que $\mathscr{C} \subseteq \frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ es un ideal. Identificando $\mathscr{C}$ con $\psi(\mathscr{C})$.

Lema 2.3. *El código* $\mathscr{C} \subseteq \mathbb{F}_q[x]_{<n}$ *es cíclico si, y sólo si* $\forall\, c(x) \in \mathscr{C}$ *el resto,* $R(xc(x), x^n - 1) \in \mathscr{C}$.

Demostración. Supongamos que $\mathscr{C}$ es cíclico. Tomamos $c(x) \in \mathscr{C}$:
Por la definición de código cíclico, si $c(x) = c_0 + c_1x + \ldots + c_{n-1}x^{n-1} \in \mathscr{C}$ entonces, $c'(x) = c_{n-1} + c_0x + \ldots + c_{n-2}x^{n-1} \in \mathscr{C}$.

$$\begin{aligned} xc(x) &= c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} + c_{n-1}x^n \\ &= c_{n-1}(x^n - 1) + c_{n-1} + c_0x + \ldots + c_{n-2}x^{n-1}. \end{aligned}$$

Entonces, $R(xc(x), x^n - 1) = c_{n-1} + c_0x + \ldots + c_{n-2}x^{n-1}$. Es decir,

$$R(xc(x), x^n - 1) = c'(x) \in \mathscr{C}.$$

Recíprocamente, si $R(xc(x), x^n - 1) \in \mathscr{C} \Longrightarrow c_{n-1} + c_0x + \ldots + c_{n-2}x^{n-1} \in \mathscr{C}$. □

Proposición 2.4. *El código* $\mathscr{C} \subseteq \mathbb{F}_q[x]_{<n}$ *es cíclico si, y sólo si,* $\forall\, p(x) \in \mathbb{F}_q[x]$, $\forall\, c(x) \in \mathscr{C}$, $R(p(x)c(x), x^n - 1) \in \mathscr{C}$

Demostración. Consecuencia del resultado anterior. □

Teorema 2.5. *El código* $\mathscr{C} \subseteq \mathbb{F}_q[x]_{<n}$ *es cíclico si, y sólo si,* $\mathscr{C}$ *es un ideal en* $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$.

Demostración. Sabemos que la aplicación ψ definida en (2.1) es isomorfismo de espacios vectoriales. Por definición, todo código $\mathscr{C}$ es subespacio vectorial de $\mathbb{F}_q^n$. Luego, $\psi(\mathscr{C})$ es espacio vectorial en $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Además, $(\psi(\mathscr{C}), +)$ es subgrupo abeliano en $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Asimismo, $c(x)p(x) \in \psi(\mathscr{C})$, $\forall\, p(x) \in \mathbb{F}_q[x]$, $\forall\, c(x) \in \mathscr{C}$, por la Proposición 2.4. Concluyendo así que $\mathscr{C}$ es un ideal en $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$.

Recíprocamente, consideremos I ideal de $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$. Como I es un subespacio de $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$, entonces $\psi^{-1}(I)$ es subespacio vectorial de $\mathbb{F}_q^n$. Dicho de otra forma, $\psi^{-1}(I)$ es un código lineal en $\mathbb{F}_q^n$. Faltaría comprobar que es cíclico. En efecto, $\forall (a_0, \ldots, a_{n-1}) \in \psi^{-1}(I)$ el polinomio $a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \in I$. Por hipótesis, I es ideal:

$$x(a_0 + a_1x + \cdots + a_{n-1}x^{n-1}) = a_{n-1}(x^n - 1) + (a_{n-1} + a_0x + \ldots + a_{n-2}x^{n-1}) \in I,$$

por tanto, $(a_{n-1}, a_0, \ldots, a_{n-2}) \in \mathscr{C}$. Concluyendo así que $\mathscr{C}$ es cíclico. □

El anillo $\mathbb{F}_q[X]$ es un dominio de ideales principales, esto significa que todos sus ideales están generados por un elemento. Además si el ideal es distinto de cero, entonces su elemento generador es único salvo múltiplos escalares de él. Por tanto, existe un único polinomio mónico que genera dicho ideal. Además

sabemos que $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ es un anillo cociente de $\mathbb{F}_q[x]$, es decir, que éste también es un dominio de ideales principales. Por el Teorema 2.5 sabemos que un código cíclico es un ideal de $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ por tanto, debe estar generado por un elemento. Sin embargo, aunque no podemos asegurar que sea único, si podemos asegurar que existe un único polinomio mónico y mínimo.

Teorema 2.6 (Teorema fundamental de códigos cíclicos). *Sea $\mathscr{C}$ un código cíclico, existe $g(x) \in \mathscr{C}$ con las siguientes propiedades:*

1. *$g(x)$ es el único polinomio mónico de grado mínimo en $\mathscr{C}$.*
2. *$\mathscr{C} = \langle g(x)\rangle$*
3. *$g(x)$ es un divisor de $x^n - 1$.*

A este polinomio $g(x)$ lo llamamos polinomio generador de $\mathscr{C}$.

Demostración. 1. Procedemos por reducción al absurdo.
Supongamos que existen $g_1(x)$, $g_2(x)$ mónicos diferentes de grado mínimo en $\mathscr{C}$, $g_1(x) \neq g_2(x) \in \mathscr{C} \setminus \{0\}$. Por tanto,

$$\deg(g_1(x) - g_2(x)) < \deg(g_1(x)) = \deg(g_2(x))$$

que contradice la minimalidad de $g_1(x)$, $g_2(x)$.

2. Para cualquier $c \in \mathscr{C}$ por la división euclídea existen $h(x)$, $r(x)$ tales que

$$c(x) = g(x)h(x) + r(x),$$

donde $\deg(r(x)) < \deg(g(x))$ o $r(x) = 0$. Si $r(x) \neq 0$, entonces, se obtiene que $r(x) = c(x) - g(x)h(x) \in \mathscr{C}$ contradiciendo la minimalidad de $g(x)$.

3. El código $\mathscr{C}$ es un ideal de $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$ y $0 \in \mathscr{C}$. Por tanto, $x^n - 1 \in \mathscr{C}$. Por apartado 2, $x^n - 1 = g(x)h(x)$ concluyendo que $g(x)$ es un divisor de $x^n - 1$.

□

Existen tantos códigos cíclicos distintos sobre $\mathbb{F}_q^n$ como divisores mónicos tenga el polinomio $x^n - 1$ en $\mathbb{F}_q[x]$.

Proposición 2.7. *Sea $g(x)$ el polinomio generador de un código cíclico $\mathscr{C}$. Entonces, $\mathscr{C}$ tiene dimensión $k = n - \deg(g(x))$.*

Demostración. Sea $\mathscr{C}$ un código cíclico entonces $\mathscr{C} = \langle g(x)\rangle$ donde $g(x)$ divisor de $x^n - 1$. Comprobemos que

$$\mathscr{C} = \{a(x)g(x) : a(x) \in \mathbb{F}_q[x] \text{ con } \deg(a(x)) < n - \deg(g(x))\}.$$

Por un lado, $\forall c \in \mathscr{C}$, $\exists\, f(x), h(x) \in \mathbb{F}_q[x]$: $c(x) = f(x)g(x) + h(x)(x^n - 1)$.
Por tanto,

$$a(x) = f(x) + h(x)\frac{x^n - 1}{g(x)} = \frac{c(x)}{g(x)}.$$

Entonces, $\deg(a(x)) = \deg(c(x)) - \deg(g(x)) < n - \deg(g(x))$.
Por otro lado, veamos que $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ es una base del código $\mathscr{C}$ con $k = n - \deg(g(x))$. Por hipótesis, $\forall c \in \mathscr{C}$, $c(x) = a(x)g(x)$ con $\deg(a(x)) < k$ es decir, $a(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$. Entonces,

$$c(x) = a(x)g(x) = a_0g(x) + a_1xg(x) + \ldots + a_{k-1}x^{k-1}g(x).$$

Es decir, $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$ es un sistema generador del código $\mathscr{C}$. Además, como todos sus elementos tienen diferente grado entonces es linealmente independiente, es decir, es una base.

□

Definición 2.8 (Matriz generatriz). *Sea $\mathscr{C}$ un $[n,k]_q$ código cíclico de longitud n con polinomio generador $g(x) = g_0 + g_1x + \ldots + g_{n-k}x^{n-k}$. Definimos la matriz generatriz $\mathcal{G}$ de $\mathscr{C}$ como,*

$$\mathcal{G} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & g_2 & \cdots & g_{n-k} \end{pmatrix}.$$

Definición 2.9 (Polinomio de paridad). *Definimos el polinomio de paridad de $\mathscr{C} = \langle g(x) \rangle$ como*

$$h(x) = \frac{x^n - 1}{g(x)} = h_0 + h_1x + \ldots + h_kx^k.$$

Definición 2.10 (Matriz de paridad). *Sea $\mathscr{C}$ un $[n,k]_q$ código cíclico con polinomio de paridad $h(x) = h_0 + h_1x + \ldots + h_kx^k$.. Definimos la matriz de paridad $\mathcal{H}$ de $\mathscr{C}$ como,*

$$\mathcal{H} = \begin{pmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_k & h_{k-1} & \cdots & h_0 \end{pmatrix}.$$

Observación 2.11. Es fácil comprobar $\mathcal{H}\mathcal{G}^T = 0$, con esto, es claro que $\mathcal{H}$ es matriz de paridad para $\mathscr{C}$.

Definición 2.12 (Código dual de un código cíclico). *Sea $\mathscr{C}$ un $[n,k]_q$ código cíclico con polinomio de control $h(x) = h_0 + h_1x + \ldots + h_kx^k$. Entonces, $\mathscr{C}^\perp$ es un código cíclico generado por*

$$\bar{h}(x) = x^k h\left(\frac{1}{x}\right) = h_0x^k + \ldots + h_{k-1}x + h_k.$$

Veamos un ejemplo en SageMath de códigos cíclicos.

Ejemplo 2.13. Sabemos que existen tantos códigos cíclicos distintos sobre $\mathbb{F}_q^n$ como divisores mónicos tenga el polinomio $x^n - 1$. En este ejemplo trabajaremos en $\mathbb{F}_2^7$, por tanto, calcularemos los divisores mónicos de $x^7 - 1$. Esto es,

$$x^7 - 1 = (x-1)(x^3 + x + 1)(x^3 + x^2 + 1) \in \mathbb{F}_2[x]$$

Por tanto, como $x^7 - 1$ tiene 3 factores irreducibles en $\mathbb{F}_2[x]$ existirán $2^3 = 8$ códigos cíclicos incluyendo al $\{0\}$ y $\mathbb{F}_2^7$.

1. Código $\mathscr{C}_1$ generado por el polinomio $g_1(x) = x - 1$. Será un código con $\dim(\mathscr{C}_1) = n - \deg(g_1(x)) = 7 - 1 = 6$.

```
Sage:F.<x> = GF(2)[]
Sage: n = 7; g = x + 1
Sage: C1 = codes.CyclicCode(generator_pol = g, length = n); C1
>[7, 6] Cyclic Code over GF(2)
// Creamos la matriz generatriz
Sage:C1.generator_matrix()
>[1 1 0 0 0 0 0]
 [0 1 1 0 0 0 0]
 [0 0 1 1 0 0 0]
 [0 0 0 1 1 0 0]
 [0 0 0 0 1 1 0]
 [0 0 0 0 0 1 1]
 // Creamos el polinomio de paridad
Sage: h = C1.check_polynomial();h
>x^6 + x^5 + x^4 + x^3 + x^2 + x + 1
Sage: h == (x**n - 1)/C1.generator_polynomial()
>True
// Creamos una matriz de paridad
Sage: C1.parity_check_matrix()
>[1 1 1 1 1 1 1]
```

2. Código $\mathscr{C}_2$ generado por el polinomio $g_1(x) = x^3 + x + 1$. Será un código con $\dim(\mathscr{C}_2) = n - \deg(g_2(x)) = 7 - 3 = 4$. Generaremos la matriz generatriz $\mathcal{G}_2 \in \mathbb{F}_2^{4\times 7}$, el polinomio de paridad $h_2(x)$ y la matriz de paridad $\mathcal{H}_2 \in \mathbb{F}_2^{3\times 7}$ para $\mathscr{C}_2$.

```
Sage: F.<x> = GF(2)[]
Sage: n = 7; g = x^3 + x + 1
Sage: C2 = codes.CyclicCode(generator_pol = g, length = n); C2
>[7, 4] Cyclic Code over GF(2)
Sage:C2.generator_matrix()
```

```
>[1 1 0 1 0 0 0]
 [0 1 1 0 1 0 0]
 [0 0 1 1 0 1 0]
 [0 0 0 1 1 0 1]
Sage: h = C2.check_polynomial();h
>x^6 + x^5 + x^4 + x^3 + x^2 + x + 1
Sage: C2.parity_check_matrix()
>[1 0 1 1 1 0 0]
 [0 1 0 1 1 1 0]
 [0 0 1 0 1 1 1]
```

3. De la misma forma se pueden calcular los códigos a partir de los polinomios generadores $g_3(x) = x^3 + x^2 + 1$, $g_4(x) = (x+1)(x^3 + x + 1)$ y $g_5(x) = (x+1)(x^3 + x^2 + 1)$.
4. Veamos el código $\mathscr{C}_6$ generado por el polinomio $g_6(x) = (x^3 + x + 1)(x^3 + x^2 + 1)$. Será un código con $\dim(\mathscr{C}_6) = n - \deg(g_1(x)) = 7 - 6 = 1$, veremos que es el código dual de $\mathscr{C}_1$, es decir, $g_6(x) = x^6 h(\frac{1}{x})$.

   ```
   Sage: F.<x> = GF(2)[]
   Sage: n = 7, g = (x^3 + x + 1)*(x^3 + x^2 + 1)
   Sage: C6 = codes.CyclicCode(generator_pol = g, length = n); C6
   >[7, 4] Cyclic Code over GF(2)
   Sage: C6.generator_polynomial()==x^6*h(1/x)
   >True
   ```

   Por tanto, $\mathscr{C}_6 = \mathscr{C}_1^{\perp}$.
5. Veamos el código $\mathscr{C}_6$ generado por el polinomio $g_7(x) = x^7 - 1$.

   ```
   Sage: F.<x> = GF(2)[]
   Sage: n = 7, g = x^7-1
   Sage: C7 = codes.CyclicCode(generator_pol = g, length = n); C7
   >[7, 0] Cyclic Code over GF(2)
   ```

   Es decir, $\mathscr{C}_7 = \{0\}$.

2.2. Conjunto de ceros de un código cíclico

Antes de trabajar con clases ciclotómicas haremos un breve resumen con los principales resultados sobre cuerpos finitos y grupos que se necesitan a lo largo de la sección. Este resumen nos servirá para fijar la denotación.

2.2.1. Cuerpos finitos

Definimos la característica de un cuerpo $\mathrm{car}(\mathbb{F})$ como el menor natural $n \in \mathbb{N}$ tal que la suma $1 + \overset{n}{\ldots} + 1 = 0$.

Observación 2.14. Sea $\mathbb{F}$ un cuerpo y $\mathrm{P}_{\mathbb{F}}$ el menor subcuerpo contenido en $\mathbb{F}$ llamado subcuerpo primo de $\mathbb{F}$. Entonces:

a. $\mathrm{car}(\mathbb{F}) = p$ con p primo si, y sólo si, $\mathrm{P}_{\mathbb{F}} \simeq \mathbb{Z}_p$.
b. $\mathrm{car}(\mathbb{F}) = 0$ si, y sólo si, $\mathrm{P}_{\mathbb{F}} \simeq \mathbb{Q}$.

Proposición 2.15. *Sea $\mathbb{F}_q$ un cuerpo finito con q elementos. Entonces:*

i. $\exists p$ *primo* : $\mathbb{F}_p \subseteq \mathbb{F}_q$.
ii. $\exists p$ *primo* : $q = p^n$ *con* $n \geq 1$.
iii. $\forall \alpha \in \mathbb{F}_q$ *entonces,* $\alpha^q = \alpha$.

Definición 2.16 (Clausura algebraica). *Sea $\mathbb{F}$ un cuerpo definimos la clausura algebraica de $\mathbb{F}$ como el menor cuerpo $\overline{\mathbb{F}}$ tal que $\mathbb{F} \subseteq \overline{\mathbb{F}}$ y donde todo polinomio con coeficientes en $\mathbb{F}$ tiene sus raíces en $\overline{\mathbb{F}}$.*

La clausura existe y es única salvo isomorfismos.

Proposición 2.17. *Sea A un dominio de integridad con* $\mathrm{car}(A) = p$. *Entonces,* $\forall \alpha, \beta \in A$: $(\alpha + \beta)^p = \alpha^p + \beta^p$.

Corolario 2.18. *Sea A un dominio de integridad con* $\mathrm{car}(A) = q = p^r$. *Entonces* $(a + b)^q = a^q + b^q$.

Teorema 2.19. *Para cualesquiera p número primo y $n \in \mathbb{N}$ existe $\mathbb{F}_q$ un cuerpo finito con $q = p^n$ elementos. Además, este cuerpo es único salvo isomorfismos.*

Construcción explícita de Cuerpos Finitos

Hemos demostrado la existencia de un cuerpo finito con q elementos $\mathbb{F}_q$. Veamos ahora cómo construirlo explícitamente. Sean $f(x) = x^n + a_{n-1}x^{n-1} + \ldots + a_0 \in \mathbb{F}_p[x]$ un polinomio mónico irreducible y α raíz de $f(x)$ consideramos el cuerpo irreducible

$$F_q = \frac{\mathbb{F}_p[x]}{\langle f(x) \rangle} = \mathbb{F}_p[\alpha] = \{a_{n-1}\alpha^{n-1} + \ldots + a_0 \mid a_0, \ldots, a_{n-1} \in \mathbb{F}_p\}$$

entonces, $\sharp \mathbb{F}_p[\alpha] = p^m$.

Definición 2.20 (Grupo cíclico, orden y exponente). *Sea* G *un grupo finito:*

- *Diremos que* G *es cíclico si* $\exists g \in \mathrm{G}$: $\mathrm{G} = \langle g \rangle$.
- *Definimos el orden de* $g \in \mathrm{G}$ *como el cardinal del subgrupo generado por g es decir,*
$$\circ(g) = \sharp \langle g \rangle = \sharp\{a, a^2, \ldots, a^n = 1\} = min\{n \mid a^n = 1\}.$$
- *Definimos el orden de* G *como el cardinal del conjunto* $\sharp \mathrm{G}$.

- *Definimos el exponente de* G *como* $\exp(\mathrm{G}) = \mathrm{mcm}\{\circ(g) \mid g \in \mathrm{G}\}$.

Observación 2.21. Para todo $g \in \mathrm{G}$ se cumple que $\frac{\circ(g)}{\sharp \mathrm{G}}$. Además, si $\sharp\mathrm{G}$ es primo entonces G es cíclico.

Lema 2.22. *Sea* G *un grupo finito de exponente* n*. Entonces,* $\exists g \in \mathrm{G}\colon \circ(g) = n$.

Teorema 2.23. *El grupo multiplicativo* $(\mathbb{F}_q^*, x)$ *es cíclico de orden* $q-1$.

Definición 2.24 (Elemento primitivo). *Definimos un elemento primitivo de* $\mathbb{F}_q$ *como un generador del grupo cíclico* $(\mathbb{F}_q^*, x)$.

Definición 2.25 (Función de Euler). *Definimos la función de Euler* $\forall n \in \mathbb{N}$ *como*

$$\varphi(n) = \sharp\{a\colon 0 < a < n \ y \ \mathrm{mcd}(a,n) = 1\}.$$

Observación 2.26. La función de Euler verifica las siguientes propiedades:

i. Si p es primo entonces $\varphi(p) = p-1$.
ii. Si p es primo y $n \in \mathbb{N}$ entonces $\varphi(p^n) = p^n - p^{n-1} = p^{n-1}(p-1)$.
iii. Sean $m, n \in \mathbb{N}$: $(m,n) = 1$ entonces $\varphi(mn) = \varphi(m)\varphi(n)$.
iv. Sea $n = p_1^{r_1} \cdots p_s^{r_s}$ la descomposición en números primos de $n \in \mathbb{N}$ entonces

$$\varphi(n) = n \prod_i (1 - \frac{1}{p_i}).$$

Además el número de elementos primitivos del grupo cíclico $\mathbb{F}_q^*$ es $\varphi(q-1)$.

2.2.2. Clases ciclotómicas

Definición 2.27 (Raíces de la unidad). *Sea* $\mathbb{F}$ *un cuerpo llamamos grupo de las raíces n-ésimas de la unidad al grupo*

$$\mathcal{U}_n(\mathbb{F}) = \{\alpha \in \overline{\mathbb{F}} \mid \alpha^n = 1\}.$$

Observación 2.28. Se demuestran fácilmente las siguientes afirmaciones:

1. Todo subgrupo finito del grupo multiplicativo $\mathbb{F} \setminus \{0\}$ es cíclico.
2. Sea $G = \langle g \rangle$ un grupo. Entonces, g^k es también generador de G si, y sólo si, el máximo común divisor $\mathrm{mcd}(\sharp G, k) = 1$.
3. Todo grupo cíclico de orden $n \in \mathbb{N}$ finito tiene $\varphi(n)$ generadores, donde φ define la función de Euler, es decir, $\varphi(n) = \sharp\{k \in \mathbb{N}\colon k \leq n, \mathrm{mcd}(k,n) = 1\}$. Teniendo en cuenta estos tres apartados:
4. El grupo de las raíces n-ésimas de la unidad $\mathcal{U}_n(\mathbb{F})$ es cíclico finito con $\sharp\mathcal{U}_n(\mathbb{F}) = n$ y $\varphi(n)$ generadores.

Definición 2.29. *Se define como raíz n-ésima primitiva de la unidad a cualquier generador del grupo* $\mathcal{U}_n(\mathbb{F})$.

Definición 2.30. *Sea* α *una n-ésima raíz primitiva de la unidad en la extensión de cuerpo* $\mathbb{F}_{q^t}$. *Definimos el polinomio mínimo de* α^i, *denotado por* $m_{\alpha^i}^{\mathbb{F}_{q^t}}(x)$, *como el polinomio mónico en* $\mathbb{F}_q[x]$ *de menor grado tal que* $m_i(\alpha^i) = 0$.

Definición 2.31 (Polinomio ciclotómico). *Sea* α *raíz n-ésima primitiva de la unidad definimos el n-ésimo polinomio ciclotómico como*

$$\Phi_n(x) = \prod_{\alpha}(x - \alpha) = \prod_{\mathrm{mcd}(s,n)=1}(x - \alpha^s).$$

Por la Observación 2.28 (4), es claro que, $\deg(\Phi_n(x)) = \varphi(n)$.

Observación 2.32. Sea α un elemento de orden n y $\beta = \alpha^i$ para $0 \leq i < n$, entonces β es un elemento de orden n si, y sólo si, $\mathrm{mcd}(i, n) = 1$.

Proposición 2.33. *Sea* Φ_d *el d-ésimo polinomio ciclotómico entonces*

$$x^n - 1 = \prod_{d/n} \Phi_d(x).$$

Demostración. Por Observación 2.32, el elemento α^i tiene orden n si, y sólo si, $\mathrm{mcd}(i, n) = 1$. Por otro lado, si β es un elemento de orden n entonces β es un cero del polinomio $x^n - 1$ además, $x^n - 1 = \prod_{0 \leq i < n}(x - \alpha^i)$. Entonces, $\beta = \alpha^i$ para algún i con $0 \leq i < n$ y $\mathrm{mcd}(i, n) = 1$. Por tanto, Φ_n tiene tantos ceros como elementos de orden n. De donde se deduce que:

$$x^n - 1 = \prod_{0 \leq i < n}(x - \alpha^i) = \prod_{d/n} \prod_{\substack{\mathrm{mcd}(i,d)=1 \\ 0 \leq i < n}} (x - \alpha^i) = \prod_{d/n} \Phi_d(x)$$

□

Del resultado anterior obtenemos la fórmula recursiva que nos permite calcular de forma recurrente la expresión explícita del n-ésimo polinomio ciclotómico,

$$\Phi_n(x) = \frac{x^n - 1}{\prod_{\substack{d/n \\ d \neq n}} \Phi_d}. \tag{2.2}$$

Proposición 2.34. *Sea* p *primo, entonces* $\Phi_p(x) = x^{p-1} + x^{p-2} + \ldots + x + 1$.

Demostración. Según la ecuación (2.2):

$$\Phi_p(x) = \frac{x^p - 1}{\Phi_1} = \frac{x^p - 1}{x - 1} = x^{p-1} + x^{p-2} + \ldots + x + 1.$$

□

Proposición 2.35. *Sea n un entero positivo tal que* $\mathrm{mcd}(q,n)=1$ *y d el menor entero positivo que verifica que* $q^d \equiv 1 \mod n$. *Entonces, $\Phi_n(x)$ se fatoriza en $\mathbb{F}_q[x]$ como producto de $\frac{\varphi(n)}{d}$ factores.*

Demostración. Sea $f(x)$ un factor irreducible de $\Phi_n(x)$ y α una raíz n-ésima de la unidad en una extensión de $\mathbb{F}_q$: $\mathbb{F}_{q^t}$. Si α es raíz de $f(x)$. Entonces,

$$\alpha \in \mathbb{F}_{q^t} \iff \alpha^{q^t} = \alpha \iff \alpha^{q^t-1} = 1 \iff q^t \equiv 1 \mod n.$$

Por tanto, $\alpha \in \mathbb{F}_{q^d}$ y α no pertenece a ningún subcuerpo propio de $\mathbb{F}_{q^d}$. Es decir, $\deg(m_\alpha^{\mathbb{F}_q}(x)) = \deg(f(x)) = d$. Como $\deg(\Phi_n(x)) = \varphi(n)$ entonces, el número de factores irreducibles coincide con $\frac{\varphi(n)}{d}$. □

Proposición 2.36. *Sea $f(x) \in \mathbb{F}_q[x]$ y β raíz de $f(x)$. Entonces, β^q es también raíz de $f(x)$.*

Demostración. Sea $f(x) = f_0 + f_1 x + \ldots + f_m x^m \in \mathbb{F}_q[x]$ y β raíz de $f(x)$. Como $a^q = a$, $\forall a \in \mathbb{F}_q$ y, además, $f(\beta) = 0$, entonces:

$$0 = (f(\beta)^q) = (f_0 + f_1\beta + \ldots + f_m\beta^m)^q = f_0 + f_1\beta^q + \ldots + f_m\beta^{mq} = f(\beta^q).$$

La tercera igualdad es consecuencia del Corolario 2.18. Concluimos, por tanto, que β^q es raíz de $f(x)$. □

Proposición 2.37. *Sea $\beta \in \mathbb{F}_q$ con $q = p^m$. Entonces, β y β^p tienen el mismo polinomio mínimo.*

Demostración. Por el Corolario 2.18 se tiene que, $f(\beta^p) = (f(\beta))^p = 0$. □

Observación 2.38. En particular, $m_\alpha^{\mathbb{F}_q}(x) = m_{\alpha^q}^{\mathbb{F}_q}(x)$.

Definición 2.39 (Clase ciclotómica). *Se define la clase ciclotómica de q módulo n que contiene al subconjunto de índices $I \subseteq \{1,\ldots,n\}$ como:*

$$\mathcal{C}_q(I) = \{iq^j \mod n \mid i \in I,\ j = 0,1,\ldots\}.$$

Si $I = \{i\}$. Entonces, utilizamos la denotación $\mathcal{C}_q(I) = \mathcal{C}_q(i)$.

Proposición 2.40. *Si* $\mathrm{mcd}(q,n) = 1$. *Entonces, las clases ciclotómicas $\mathcal{C}_q(i)$ generan una partición de $\mathbb{Z}_n$.*

Demostración. Denotaremos el conjunto de los enteros no negativos como $\mathbb{N}_0$. Sabemos que cada $i \in \mathbb{Z}_n$ pertenece a una clase ciclotómica $\mathcal{C}_q(i)$. Supondremos que las clases $\mathcal{C}_q(i)$ y $\mathcal{C}_q(j)$ tienen elementos en común. Esto es, $iq^k = jq^l$ para ciertos $k,l \in \mathbb{N}_0$. Asumimos que $k \leq l$ por lo que $i = jq^{l-k}$ con $l-k \in \mathbb{N}_0$. Entonces, $iq^m = jq^{l-k+m}$, $\forall m \in \mathbb{N}_0$. Por tanto, $\mathcal{C}_q(i)$ está contenido en $\mathcal{C}_q(j)$. Por hipótesis, sabemos que n y q son coprimos por lo que q es invertible en $\mathbb{Z}_n$ y $q^e \equiv 1 \mod n$ para cierto $e \in \mathbb{N}_0$. Entonces, $jq^m = iq^{(e-1)(l-k)+m}$, $\forall m \in \mathbb{N}_0$. Por tanto, $\mathcal{C}_q(j)$ está contenido en $\mathcal{C}_q(i)$. Concluyendo así que dos clases ciclotómicas son iguales o disjuntas. □

Ejemplo 2.41. Veamos todas las clases ciclotómicas de 2 módulo 15, como $\text{mcd}(2, 15) = 1$ formarán una partición de $\mathbb{Z}_{15}$.

Clase ciclotómica	Elementos de la clase	Clases equivalentes
$\mathcal{C}_2(0)$	$\{0\}$	–
$\mathcal{C}_2(1)$	$\{1, 2, 4, 8\}$	$\mathcal{C}_2(2)$, $\mathcal{C}_2(4)$, $\mathcal{C}_2(8)$
$\mathcal{C}_2(3)$	$\{3, 6, 9, 12\}$	$\mathcal{C}_2(6)$, $\mathcal{C}_2(9)$, $\mathcal{C}_2(12)$
$\mathcal{C}_2(5)$	$\{5, 10\}$	$\mathcal{C}_2(5)$, $\mathcal{C}_2(10)$
$\mathcal{C}_2(7)$	$\{7, 11, 13, 14\}$	$\mathcal{C}_2(11)$, $\mathcal{C}_2(13)$, $\mathcal{C}_2(14)$

Proposición 2.42. *Sea n un entero positivo tal que $\text{mcd}(n, q) = 1$. Entonces, los posibles polinomios mínimos que generen un código $m_i^{\mathbb{F}_q}(x)$ equivalen al número de factores irreducibles de $\Phi_n(x)$ que es igual a $\frac{\varphi(n)}{d}$ con $d = \sharp\mathcal{C}_q(1)$.*

Demostración. Sea $i \in \mathbb{Z}_n$ con $\text{mcd}(i, n) = 1$ y consideremos la aplicación $\mathcal{C}_q(1) \longrightarrow \mathcal{C}_q(i)$ definida por $j \longmapsto ij$. Esta aplicación está bien definida y tiene aplicación inversa, por lo que, i es invertible en $\mathbb{Z}_n$. Por tanto, $\sharp\mathcal{C}_q(1) = \sharp\mathcal{C}_q(i)$. Además, por la Proposición 2.40 el conjunto de los elementos de $\mathbb{Z}_n$ tales que $\text{mcd}(i, n) = 1$ particiona $\mathbb{Z}_n$ en clases ciclotómicas de tamaño d y cada elección de una clase corresponde con una elección de $m_i^{\mathbb{F}_q}(x)$ que son factores mónicos irreducibles de $\Phi_n(x)$. Por tanto, el número de polinomios mínimos posibles $m_i^{\mathbb{F}_q}(x)$ es $\frac{\varphi(n)}{d}$. □

Observación 2.43. Dos propiedades interesantes para el polinomio mínimo son: Sea $m_{\alpha^i}^{\mathbb{F}_q}(x)$ el polinomio mínimo de α^i, con α raíz primitiva de la unidad en $\mathbb{F}_q$. Entonces,

$$m_{\alpha^i}(x) = \prod_{j \in \mathcal{C}_q(i)} (x - \alpha^j).$$

Sean $i, j \in \mathbb{Z}$ con $0 \leq i, j \leq n$ tales que $ij \equiv 1 \mod n$. Entonces, $m_i(x) = \text{mcd}(m_1(x^j), x^n - 1)$.

Véanse las demostraciones [1, Proposition 4.2.46] y [1, Proposition 4.2.50] respectivamente.

Ejemplo 2.44. Teniendo en cuenta el ejemplo 2.41, calculemos los polinomios mínimos.

Clase ciclotómica	Polinomio mínimo
$\mathcal{C}_2(0)$	$m_{\alpha^0}(x) = x$
$\mathcal{C}_2(1)$	$m_{\alpha^1}(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4)(x - \alpha^8)$
$\mathcal{C}_2(3)$	$m_{\alpha^3}(x) = (x - \alpha^3)(x - \alpha^6)(x - \alpha^9)(x - \alpha^{12})$
$\mathcal{C}_2(5)$	$m_{\alpha^5}(x) = (x - \alpha^5)(x - \alpha^{10})$
$\mathcal{C}_2(7)$	$m_{\alpha^7(x)} = (x - \alpha^7)(x - \alpha^{11})(x - \alpha^{13})(x - \alpha^{14})$

Definición 2.45 (Conjunto de ceros de un código cíclico). *Sean $\mathscr{C}$ un $[n,k]_q$ código cíclico, $g(x)$ el polinomio generador de $\mathscr{C}$ y α un elemento de orden n en una extensión $\mathbb{F}_{q^m}$ de $\mathbb{F}_q$. Se define el conjunto de ceros de $\mathscr{C}$ como:*

$$\mathcal{Z}(\mathscr{C}) = \{i \in \mathbb{Z}_n \colon g(\alpha^i) = 0\}.$$

Observación 2.46. La relación entre el polinomio generador de un código cíclico $\mathscr{C}$ y el conjunto de ceros $\mathcal{Z}(\mathscr{C})$ viene dada por:

$$g(x) = \prod_{i \in \mathcal{Z}(\mathscr{C})} (x - \alpha^i)$$

con $\dim(\mathscr{C}) = n - \sharp\mathcal{Z}(\mathscr{C})$.

Proposición 2.47. *El conjunto de ceros de $\mathscr{C}$ coincide con la unión disjunta de clases ciclotómicas. Es decir,*

$$\mathcal{Z}(\mathscr{C}) = \dot{\bigcup} \mathcal{C}_q(i).$$

Demostración. Sea $g(x)$ el polinomio generador de un código cíclico $\mathscr{C}$, por la Observación 2.46, $g(\alpha^i) = 0$ si, y sólo si, $i \in \mathcal{Z}(\mathscr{C})$. Además, por la Observación 2.38, si α^i es un cero de $g(x)$ entonces, $\alpha^i q$ es un cero de $g(x)$. Por lo que, $\mathcal{C}_q(i)$ está contenido en $\mathcal{Z}(\mathscr{C})$ si $i \in \mathcal{Z}(\mathscr{C})$. Por tanto, $\mathcal{Z}(\mathscr{C})$ es unión de clases ciclotómicas. Además, esta unión es disjunta por la Proposición 2.40. □

Ejemplo 2.48. Continuando con el ejemplo 2.41 vamos a considerar el código $\mathscr{C}$ generado por $g(x) = m_{\alpha^1}(x)m_{\alpha^3}(x) = (1 + x + x^4)(1 + x + x^2 + x^3 + x^4)$. El código $\mathscr{C}$ tiene parámetros $[2^4 - 1, n - \deg(g(x))] = [15, 15 - 8] = [15, 7]$. Por la Proposición 2.47 sabemos que,

$$\mathcal{Z}(\mathscr{C}) = \{1, 2, 3, 4, 6, 8, 9, 12\}.$$

SageMath nos permite calcular el número de ceros a partir del polinomio generador de un código cíclico, por tanto, comprobamos lo anteriormente expuesto:

```
Sage: R.<x> = F[]
Sage: n = 15
Sage: g = (1+x+x^4)*(1+x+x^2+x^3+x^4)
Sage: C = codes.CyclicCode(generator_pol=g, length=n)
Sage: C.defining_set()
>[1, 2, 3, 4, 6, 8, 9, 12]
```

Si conocemos el conjunto de ceros de un código cíclico, el siguiente resultado nos permite conocer los ceros de su código dual.

Proposición 2.49. *[1, Proposition 4.3.8]. Sea $\mathscr{C}$ un código cíclico de longitud n. Entonces:*

$$\mathcal{Z}(\mathscr{C}^{\perp}) = \mathbb{Z}_n \setminus \{-i \mid i \in \mathcal{Z}(\mathscr{C})\}$$

Proposición 2.50. *Sea $N_q(n)$ el número de clases ciclotómicas de $\mathcal{C}_q(i)$ módulo n con respecto de q. Entonces, el número de códigos cíclicos de longitud n en $\mathbb{F}_q$ es $2^{N_q(n)}$.*

Demostración. Un código cíclico $\mathscr{C}$ de longitud n sobre $\mathbb{F}_q$ está determinado por su conjunto de ceros $\mathcal{Z}(\mathscr{C})$ por la Observación 2.46. El conjunto de ceros es unión disjunta de clases ciclotómicas $\mathcal{C}_q(i)$ módulo n con respecto de q por la Proposición 2.47. Por tanto, un código cíclico esta determinado por la elección de un subconjunto de todas las $N_q(n)$ clases ciclotómicas y hay $2^{N_q(n)}$ subconjuntos.

2.2.3. Códigos BCH

Una familia particular de códigos cíclicos es la llamada códigos **BCH**, por las siglas de los investigadores que los descubrieron Raj Bose, D. K. Ray-Chaudhuri y Alexis Hocquenghem. Se definen como los códigos cíclicos con una distancia mínima esperada.

Definición 2.51. *Sean $\mathscr{C}$ un código lineal en $\mathbb{F}_q^n$ y $\hat{\mathscr{C}}$ un código lineal en $\mathbb{F}_{q^m}^n$.*

- *Si $\mathscr{C} \subseteq \hat{\mathscr{C}} \cap \mathbb{F}_q^n$. Diremos que $\mathscr{C}$ es un subcódigo de $\hat{\mathscr{C}}$ en $\mathbb{F}_q^n$ y que $\hat{\mathscr{C}}$ es un súper-código de $\mathscr{C}$.*
- *Si $\mathscr{C} = \hat{\mathscr{C}} \cap \mathbb{F}_q^n$. Diremos que $\mathscr{C}$ es la restricción de $\hat{\mathscr{C}}$ en $\mathbb{F}_q^n$ y, por tanto, $\mathrm{d}_{min}(\hat{\mathscr{C}}) \leq \mathrm{d}_{min}(\mathscr{C})$.*

Teorema 2.52 (Cota BCH para la d_{min} de códigos cíclicos). *Sea $\mathscr{C}$ un $[n,k]_q$ código cíclico con $g(x)$ polinomio generador. Si el conjunto de ceros $\mathcal{Z}(\mathscr{C})$ está formado por $\delta - 1$ elementos consecutivos, es decir,*

$$\mathcal{Z}(\mathscr{C}) \supseteq \{i \colon b \leq i \leq b + \delta - 2,\, g(\beta^i) = 0\} \text{ con } \beta \in \mathbb{F}_{q^m},\, b \in \mathbb{Z}_n.$$

Entonces, $\mathrm{d}_{min}(\mathscr{C}) \geq \delta$.

Demostración. Consideremos el código lineal $\hat{\mathscr{C}}$ en $\mathbb{F}_{q^m}$ con matriz de paridad:

$$\mathcal{H} = \begin{pmatrix} 1 & \beta^b & \beta^{2b} & \cdots & \beta^{b(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{b+\delta-2} & \beta^{2(b+\delta-2)} & \cdots & \beta^{(b+\delta-2)(n-1)} \end{pmatrix} \in \mathbb{F}_{q^m}^{(\delta-1)\times n}$$

Observamos que $\mathcal{G}\mathcal{H}^T = (g(\beta^b), \cdots, g(\beta^b + \delta - 2))^T = 0$. Por tanto, $\mathscr{C} = \hat{\mathscr{C}} \cap \mathbb{F}_q^n$.

Calculamos el determinante de $(\delta - 1)$ columnas de $\mathcal{H}$ tomando cualquier subconjunto de índices $\{i_1, \cdots, i_{\delta-1}\} \subseteq \{0, \cdots, n-1\}$.

$$\det\begin{pmatrix} \beta^{bi_1} & \beta^{bi_2} & \dots & \beta^{bi_{\delta-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{(b+\delta-2)i_1} & \beta^{(b+\delta-2)i_2} & \dots & \beta^{(b+\delta-2)i_{\delta-1}} \end{pmatrix} = \det\begin{pmatrix} x_1^b & \cdots & x_{\delta-1}^b \\ x_1^{b+1} & \cdots & x_{\delta-1}^{b+1} \\ \vdots & \ddots & \vdots \\ x_1^{b+\delta-2} & \cdots & x_{\delta-1}^{b+\delta-2} \end{pmatrix} =$$

$$= (x_1^b \cdots x_{\delta-1}^b)\det\begin{pmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_{\delta-1} \\ \vdots & \ddots & \vdots \\ x_1^{\delta-2} & \cdots & x_{\delta-1}^{\delta-2} \end{pmatrix} \neq 0$$

por ser una matriz de Vandermonde. Por tanto, tenemos $\delta-1$ columnas independientes en $\mathcal{H}$. Concluyendo por el Lema 1.18 que $\delta = \mathrm{d}_{min}(\hat{\mathscr{C}}) \leq \mathrm{d}_{min}(\mathscr{C})$. □

Definición 2.53 (Código BCH). *Diremos que $\mathscr{C}$ es un código **BCH** con distancia mínima esperada δ, $\mathrm{d}_{min}(\mathscr{C}) \geq \delta$, si $\mathscr{C}$ es un código cíclico en $\mathbb{F}_q$ con*

$$\mathcal{Z}(\mathscr{C}) \supseteq \{\beta^b, \beta^{b+1}, \cdots, \beta^{b+\delta-2} \mid \beta \in \mathbb{F}_{q^m}\}.$$

*Si $n = q^m - 1$ se dice que $\mathscr{C}$ es un código **BCH** primitivo.*
*Si $b = 1$ se dice que $\mathscr{C}$ es un código **BCH** en el sentido estricto.*

Observación 2.54. Para construir un código **BCH** con distancia mínima esperada δ sobre $\mathbb{F}_q$ bastará considerar el código cíclico generado por:

$$g(x) = \mathrm{mcm}(m_b(x), m_{b+1}(x), \cdots, m_{b+\delta-2}(x)) \text{ para cierto } b \in \mathbb{Z}_n.$$

Ejemplo 2.55. Continuamos con el ejemplo 2.41, habíamos calculado que $\mathcal{Z}(\mathscr{C}) = \{1, 2, 3, 4, 6, 8, 9, 12\}$. Por la cota BCH, como $\mathcal{Z}(\mathscr{C})$ tiene $\delta - 1 = 4$ elementos consecutivos entonces $\mathrm{d}_{min}(\mathscr{C}) \geq 5$. Sagemath nos permite calcular la cota BCH.

```
Sage: F = GF(2, 'a')
Sage: n = 15
Sage: D = [1,2,3,4,6,8,9,12]
Sage: C = codes.CyclicCode(field = F, length = n, D = D)
Sage: C.bch_bound()[0]
>5
```

Por otro lado, como $\mathrm{w}_H(g(x)) = 5$ entonces, $\mathrm{d}_{min}(\mathscr{C}) \leq 5$. Por tanto, $\mathrm{d}_{min}(\mathscr{C}) = 5$. Observamos también que $\{b, b+1, b+2, b+3\} \subseteq \mathcal{Z}(\mathscr{C})$ con $b = 1$. Por tanto, $\mathscr{C}$ será un código **BCH** en el sentido estricto. Además, Sagemath nos permite crearlo a partir de su número de ceros, es decir,

```
Sage: C = codes.CyclicCode(field=GF(2),length=15,
Sage: D = [1,2,3,4,6,8,9,12])
Sage: D = codes.decoders.CyclicCodeSurroundingBCHDecoder(C)
Sage: D.bch_code()
>[15, 7] BCH Code over GF(2) with designed distance 5
```

2.2.4. Códigos Reed-Solomon

Otra familia particular de códigos cíclicos son los códigos Reed-Solomon que abreviaremos como **RS**. Al igual que en la familia anterior su nombre procede de los investigadores que los descubrieron, en este caso, de Irving S. Reed y Gustave Solomon. Destacamos que será una subfamilia de los códigos **BCH**.

Sean α un elemento primitivo de $\mathbb{F}_q$, $n = q-1$ y b, k enteros positivos tales que $0 \leq b, k \leq n$. Definimos el polinomio

$$g_{b,k}(x) = (x - \alpha^b) \cdots (x - \alpha^{b+n-k-1})$$

Por tanto, $\mathcal{Z}(\mathscr{C}) = \{b, b+1, \cdots, b+n-k-1\}$, es decir, $\delta = n - k + 1$.

Definición 2.56 (Código Reed-Solomon). *El código Reed-Solomon* $\mathbf{RS}_k(n, b)$ *es el código cíclico con polinomio generador* $g_{b,k}(x)$.

Proposición 2.57. *Un código* $\mathbf{RS}_k(n, b)$ *es un código cíclico con los parámetros* $[n = q-1, k]_q$ *y* $\mathrm{d}_{min} = n - k + 1$, *es decir, es un código MDS.*

Demostración. Por definición, $\mathbf{RS}_k(n, b)$ es cíclico con $n = q - 1$. Sabemos que $\deg(g_{b,k}(x)) = n - k$ y por la Proposición 2.7, $\dim(\mathbf{RS}_k(n, b)) = k$. Utilizando la cota de Singleton (Proposición 1.19), $\mathrm{d}_{min}(\mathscr{C}) \leq n - k + 1$. Por otro lado, utilizando la cota **BCH**, $\mathrm{d}_{min}(\mathscr{C}) \geq \delta = n - k + 1$. Concluimos, por tanto, $\mathrm{d}_{min}(\mathscr{C}) = n - k + 1$. □

Proposición 2.58. *El código dual de* $\mathbf{RS}_k(n, b)$ *es* $\mathbf{RS}_{n-k}(n, n - b + 1)$.

Demostración. Por la Proposición 2.49:

$$\begin{aligned}
\mathcal{Z}(\mathscr{C}^\perp) &= \mathbb{Z}_n \setminus \{-i \mid i \in \mathcal{Z}(\mathscr{C})\} \\
&= \mathbb{Z}_n \setminus \{-b, -(b+1), \ldots, -(b+n-k-1)\} \\
&= \{n-b+1, n-b+2, \ldots, n-b+k\} \\
&= \{n-b+1, n-b+2, \ldots, (n-b+1)+n-(n-k)-1\}.
\end{aligned}$$

Por tanto, $(\mathbf{RS}_k(n, b))^\perp = \mathbf{RS}_{n-k}(n, n - b + 1)$. □

Ejemplo 2.59. Consideremos un código cíclico $[6, 3]_7$ con polinomio generador

$$g(x) = 6 + x + 3x^2 + x^3 = (x-2)(x-3)(x-6) = (x-\alpha)(x-\alpha^2)(x-\alpha^3),$$

donde $\alpha = 3$ elemento primitivo de $\mathbb{F}_7$. Por tanto, los ceros del código $\mathscr{C}$ serán $\mathcal{Z}(\mathscr{C}) = \{1, 2, 3\}$. Procedamos a calcular su código dual $\mathscr{C}^\perp$.

```
Sage: F.<x> = GF(7)[]
Sage: n = 6
Sage: g = (x-2)*(x-3)*(x-6)
Sage: C = codes.CyclicCode(generator_pol = g, length = n); C
Sage: h = C.check_polynomial();h
>x^3 + 4*x^2 + x + 1
```

Sabemos que el polinomio generador del código dual $\mathscr{C}^{\perp}$ es

$$\bar{h} = x^3 h(\tfrac{1}{x}) = x^3 + 4 * x^2 + x + 1 = (x-1)(x-\alpha)(x-\alpha^2).$$

Por tanto, $\mathcal{Z}(\mathscr{C}^{\perp}) = \{0,1,2\} = \mathbb{Z}_6 \setminus \{-1,-2,-3\}$ como habíamos demostrado en la Proposición 2.49. Además, observamos que $\mathscr{C} = \langle g(x) \rangle = \mathbf{RS}_3(6,1)$ por tanto por 2.58, es claro que $\mathscr{C}^{\perp} = \langle \bar{h}(x) \rangle = \mathbf{RS}_3(6,6)$

Los códigos Reed-Solomon, además de ser códigos cíclicos, pertenecen a la familia de códigos de evaluación.

Definición 2.60 (Códigos RS como códigos de evaluación). *Considerando $f(x) \in \mathbb{F}_q[x]$, definimos la aplicación evaluación como:*

$$\begin{aligned} ev\colon \mathbb{F}_q[x] &\longrightarrow \mathbb{F}_q^n \\ f(x) &\longmapsto ev(f(x)) = (f(1), f(\alpha), \ldots, f(\alpha^{n-1})) \end{aligned}$$

Otra definición de códigos Reed-Solomon es la siguiente:

$$\mathbf{RS}_k(n,b) = \{ev(x^{n-b+1} f(x)) \mid f(x) \in L_k\},$$

con $L_k = \{f(x) \in \mathbb{F}_q[x] \mid \deg(f) < k\}$.

2.2.5. Códigos Reed-Solomon generalizados

Los códigos Reed-Solomon generalizados **GRS** son un subconjunto de los códigos Reed-Solomon. Para su definición definimos la operación $*$ entre dos vectores como el producto de sus coordenadas $\mathbf{a} * \mathbf{b} = (a_1 b_1, a_2 b_2, \ldots, a_n b_n)$

Definición 2.61 (Código Reed-Solomon generalizado). *Sea $n = q-1$, $1 \leq k \leq n$ tomando $\mathbf{a} = (a_1, a_2, \cdots, a_n) \in \mathbb{F}_q^n$ con $a_i \neq a_j$, $\forall i \neq j$ y considerando $\mathbf{b} = (b_1, b_2, \cdots, b_n) \in \mathbb{F}_q^n$ con $b_i \neq 0$, definimos la evaluación:*

$$\begin{aligned} ev_{\mathbf{a},\mathbf{b}}\colon \mathbb{F}_q[x] &\longrightarrow \mathbb{F}_q^n \\ f(x) &\longmapsto \mathbf{b} * f(\mathbf{a}) = (b_1 f(a_1), b_2 f(a_2), \ldots, b_n f(a_n)) \end{aligned}$$

Por tanto, $\boldsymbol{GRS}_k(\mathbf{a},\mathbf{b}) = \{ev_{\mathbf{a},\mathbf{b}}(f) \mid f(x) \in L_k\}$.

Observación 2.62. Sean α un elemento primitivo de $\mathbb{F}_q$, $n = q-1$, $a_j = \alpha^{j-1}$ y $b_j = a_j^{n-b+1}$ con $j = 1, \ldots, n$. Se comprueba que los códigos **GRS** son códigos **RS**. En particular:

$$\mathbf{RS}_k(n,b) = \mathbf{GRS}_k(\mathbf{a},\mathbf{b})$$

Decodificación para códigos GRS

Sea $\mathbf{y} = \mathbf{c} + \mathbf{e}$ el vector recibido con $\mathbf{c} \in \mathscr{C} = \mathbf{GRS}_k(\mathbf{a}, \mathbf{b})$ y $\mathbf{e}$ el error generado durante la transmisión de la palabra. Como $\mathbf{c} \in \mathbf{GRS}_k(\mathbf{a}, \mathbf{b})$, sabemos que $\exists f \in L_k \colon \mathbf{c} = ev_{\mathbf{a},\mathbf{b}}(f) = \mathbf{b} * f(\mathbf{a})$.

1. Definimos el conjunto de las posiciones de error como:

$$I = \{i \in \{1, \ldots, n\} \mid b_i f(a_i) \neq y_i\} = \{i_1, \ldots, i_t\}.$$

   Recordemos que este conjunto es desconocido.
2. Definimos el polinomio $E(X) = \prod_{i \in I}(X - a_i)$. Y consideramos:

$$E(X) b_i f(a_i) = E(X) y_i, \; \forall i \in \{1, \ldots, n\} \tag{2.3}$$

   que cumple que:
   - Si $i \in I$ entonces, $E(a_i) = 0$.
   - En caso contrario, $b_i f(a_i) = y_i$.

   Sabemos que $E(X)$ es un polinomio de grado t. Por tanto, la parte derecha de la igualdad (2.3) se corresponde con el polinomio:

$$E(X) y_i = X^t + \sum_{i=0}^{t-1} A_i X^i$$

   donde los coeficientes $A_i \in \mathbb{F}_q$ no son conocidos. Por otro lado, $E(X) f(X)$ es un polinomio de grado menor o igual que $t + (k-1)$. Por tanto, la parte izquierda de la igualdad (2.3) se corresponde con el polinomio

$$E(X) = \sum_{i=0}^{t+k-1} B_i X^i$$

   donde los coeficientes $B_i \in \mathbb{F}_q$ tampoco son conocidos
3. Por lo tanto el siguiente sistema tiene n ecuaciones y $2t + k$ incógnitas.

$$\sum_{i=0}^{t+k-1} B_i X^i = X^t + \sum_{i=0}^{t-1} A_i X^i.$$

   Este sistema tiene solución si $2t + k < n$. Por tanto, como $\mathscr{C}$ es un código MDS y cumple, $d_{min}(\mathscr{C}) = n - k + 1$, podemos corregir, $t < \frac{n-k}{2} = \frac{\mathrm{d}_{min}(\mathscr{C})-1}{2}$.

3

Decodificación algebraica por pares correctores de errores

3.1. Pares correctores de errores

La noción de pares correctores de errores (**ECP** por su traducción del inglés "Error Correcting Pairs") fue introducida de forma independiente en 1992 por Pellikaan [4] y Kötter [5]. En este capítulo veremos que un código lineal con un par t-corrector de errores tiene un algoritmo de decodificación eficiente que permite corregir t errores con complejidad $\mathcal{O}(n^3)$ (complejidad polinomial). Con la noción de **ECP** se pueden describir diferentes algoritmos clásicos de decodificación de algunas familias de códigos algebraicos como los códigos **RS** y **BCH**.

Definición 3.1 (Producto estrella). *Sean* $\mathbf{a} \in \mathrm{A}$ *y* $\mathbf{b} \in \mathrm{B}$ *definimos la operación* $*$ *como el producto de sus coordenadas:* $\mathbf{a} * \mathbf{b} = (a_1 b_1, a_2 b_2, \ldots, a_n b_n)$. *Esta operación también se puede generalizar a subconjuntos:*

$$\mathrm{A} * \mathrm{B} = \langle \{ \mathbf{a} * \mathbf{b} \mid \mathbf{a} \in \mathrm{A}, \mathbf{b} \in \mathrm{B} \} \rangle .$$

Si $\mathrm{A} = \mathrm{B}$, *entonces* $\mathrm{A} * \mathrm{A} = \mathrm{A}^{(2)}$.

A partir de ahora denotaremos la longitud, la dimensión y la distancia mínima de un código $\mathscr{C}$ como $n(\mathscr{C})$, $k(\mathscr{C})$, $\mathrm{d}_{min}(\mathscr{C})$, respectivamente.

Observación 3.2. Es fácil comprobar que si $\mathscr{C}_1$ y $\mathscr{C}_2$ son códigos en $\mathbb{F}_q^n$ entonces:

$$\dim(\mathscr{C}_1 * \mathscr{C}_2) = \binom{k(\mathscr{C}_1) + k(\mathscr{C}_2)}{2}.$$

Definición 3.3 (Par t-corrector de errores). *Sean* $\mathscr{C}$ *un código lineal en* $\mathbb{F}_q^n$ *y* A, B, *códigos lineales en* $\mathbb{F}_{q^m}^n$. *Entonces,* (A, B) *es un par* t*-corrector de errores (*t*-ECP) para* $\mathscr{C}$ *si cumple las siguientes propiedades:*

(1) $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$
(2) $k(\mathrm{A}) > t$
(3) $\mathrm{d}_{min}(\mathrm{B}^{\perp}) > t$
(4) $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) > n$

Si un par (A, B) cumple las tres primeras condiciones (1)-(3) se dirá que es un par t-localizador de errores (**ELP** por su traducción del inglés "Error Locating Pairs").

3.1.1. Algoritmo de decodificación con ECP

Supongamos que recibimos el vector $\mathbf{y} \in \mathbb{F}_q^n$ en el que se ha cometido un cierto error, es decir $\mathbf{y} = \mathbf{c} + \mathbf{e}$ con $\mathbf{c} \in \mathscr{C}$. Además supongamos que (A, B) es un par t-corrector de errores para $\mathscr{C}$.

En primer lugar definimos los siguientes subconjuntos:

$$K_{\mathbf{y}} = \{\mathbf{a} \in \mathrm{A} \mid \langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = 0 \ , \ \forall \mathbf{b} \in \mathrm{B}\}$$

$$\mathrm{A}(J) = \{\mathbf{a} \in \mathrm{A} \mid a_j = 0 \ , \ \forall j \in J\} \text{ donde } J \subseteq \{1, \ldots, n\}$$

Lema 3.4. *Sea* $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$*, entonces* $K_y = K_e$*.*

Demostración. Sean $\mathbf{a} \in \mathrm{A}$ y $\mathbf{b} \in \mathrm{B}$ entonces,

$$\langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{c} + \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{c}, \mathbf{a} * \mathbf{b} \rangle + \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle$$

donde la tercera igualdad es consecuencia de la definición de código dual $\mathcal{C}^{\perp}$ (Def. 1.9). □

Lema 3.5. *Sean* $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$ *e* $I = \mathrm{supp}(\mathbf{e})$*, entonces* $\mathrm{A}(I) \subseteq K_{\mathbf{y}}$*. Si además se cumple que* $\mathrm{d}_{min}(\mathrm{B}^{\perp}) > t = \mathrm{w}_H(\mathbf{e})$ *entonces,* $\mathrm{A}(I) = K_{\mathbf{y}}$*.*

Demostración. Por definición, para todo $\mathbf{a} \in \mathrm{A}(I)$ se tiene que $a_j = 0$, $\forall j \in I$. Por lo tanto, $\forall \mathbf{a} \in \mathrm{A}(I)$ y $\mathbf{b} \in \mathrm{B}$ se tiene que:

$$\langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = \sum_{j=1}^{n} e_j a_j b_j = \sum_{j \in \mathrm{supp}(\mathbf{e})} e_j a_j b_j + \sum_{j \notin \mathrm{supp}(\mathbf{e})} e_j a_j b_j = 0$$

Es decir $\mathrm{A}(I) \subseteq K_{\mathbf{y}}$.

Si además se tiene que $\mathrm{w}_H(\mathbf{e}) \leq t$. Entonces $\forall \mathbf{a} \in K_{\mathbf{y}}$ se tiene que

$$0 = \langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e} * \mathbf{a}, \mathbf{b} \rangle \ , \ \forall \mathbf{b} \in \mathrm{B}$$

Es decir $\mathbf{e} * \mathbf{a} \in \mathrm{B}^{\perp}$. Sin embargo, $\mathrm{w}_H(\mathbf{e} * \mathbf{a}) \leq \mathrm{w}_H(\mathbf{e}) \leq t < \mathrm{d}_{min}(\mathrm{B}^{\perp})$, luego $\mathbf{e} * \mathbf{a} = 0$. Es decir, $K_{\mathbf{y}} \subseteq \mathrm{A}(I)$. □

Lema 3.6. *Sean* $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$, $I = \mathrm{supp}(\mathbf{e})$ *y* $k(\mathrm{A}) > t = \mathrm{w}_H(\mathbf{e})$, *entonces existe* $\mathbf{a} \in K_{\mathbf{y}} \setminus \{0\}$.

Demostración. Sabemos por el Lema 3.5 que $\mathrm{A}(I) \subseteq K_{\mathbf{y}}$. Además, $\mathrm{A}(I)$ es la intersección de t subespacios de codimensión 1. Por tanto, si $k(\mathrm{A}) > t$ entonces $\exists \mathbf{a} \neq 0 \colon\ \mathbf{a} \in \mathrm{A}(I)$, es decir, $\exists \mathbf{a} \in K_{\mathbf{y}}$. □

Sea $\mathbf{a} \in K_{\mathbf{y}} \setminus \{0\}$ definimos el conjunto $J = \{j \mid a_j = 0\} = \overline{\mathrm{supp}(\mathbf{a})}$ como el complementario del soporte de $\mathbf{a}$.

Lema 3.7. *Sea* $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$.

a. Si $\mathrm{d}_{min}(\mathrm{B}) > t$, *entonces* $I = \mathrm{supp}(\mathbf{e}) \subseteq J$.
b. Si $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) > n$, *entonces existe un único* $\mathbf{e}$ *tal que*

$$< \mathbf{e}, \mathbf{a} * \mathbf{b} >= 0, \forall \mathbf{b} \in \mathrm{B} \ \textit{con}\ I = \mathrm{supp}(\mathbf{e}) \subseteq J.$$

Demostración. a. Por el Lema 3.5 sabemos que si $\mathbf{a} \in K_{\mathbf{y}} \setminus \{0\}$ y además, $\mathrm{d}_{min}(\mathrm{B}) > t \geq \mathrm{w}_H(\mathbf{e})$, entonces $\mathbf{e} * \mathbf{a} = 0$. Es decir

$$I = \mathrm{supp}(\mathbf{e}) \subseteq \overline{\mathrm{supp}(\mathbf{a})} = J.$$

b. Supongamos que $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathrm{B}) > n$. Procedemos por reducción al absurdo. Sea $\langle \mathbf{e_1}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e}_2, \mathbf{a} * \mathbf{b} \rangle = 0$, $\forall \mathbf{b} \in \mathrm{B}$, con $\mathrm{supp}(\mathbf{e_1}), \mathrm{supp}(\mathbf{e_2}) \subseteq J$. Entonces, $\mathbf{e}_1 - \mathbf{e}_2 \in \mathscr{C}$ pero $\mathrm{w}_H(\mathbf{e_1} - \mathbf{e_2}) \leq n - |\mathrm{supp}(\mathbf{a})| \leq \mathrm{d}_{min}(\mathscr{C}) - 1$ contradiciendo la distancia mínima de $\mathscr{C}$. Por tanto, $\mathbf{e_1} - \mathbf{e_2} = 0$ es decir, $\mathbf{e_1} = \mathbf{e_2}$.

□

Decodificación con pares correctores de errores. Sea $\mathbf{y} = \mathbf{c} + \mathbf{e}$ el vector recibido donde $\mathbf{c} \in \mathscr{C}$ y $\mathbf{e}$ es el error generado durante la transmisión. Sean (A, B) un par t-corrector de errores para $\mathscr{C}$. El algoritmo para decodificar $\mathscr{C}$ que nos proporciona el **ECP** es el siguiente:

1. Encontrar $\mathbf{a} \in \mathrm{A} \colon\ \mathbf{a} \in K_{\mathbf{y}} \setminus \{0\}$. Entonces por el Lema 3.4

$$\langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = 0 \ , \forall \mathbf{b} \in \mathrm{B}$$

   Si $K_{\mathbf{y}} = 0$, por el Lema 3.6, entonces se han producido más de t errores.
2. Definimos $J = \{j \mid a_j = 0\} = \overline{\mathrm{supp}(\mathbf{a})}$.
   Sabemos por Lema 3.7 que $\mathrm{supp}(\mathbf{e}) \subseteq J$
3. Encontrar la solución única $\mathbf{e} \in \mathbb{F}_q^n$ tal que $\forall \mathbf{b} \in \mathrm{B} \colon \langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = 0$ con $\mathrm{supp}(\mathbf{e}) \subseteq J$.
   Por el Lema 3.7 sabemos que la solución al sistema anterior es única.

Puntualizamos que los dos primeros pasos del algoritmo sirven para localizar errores mientras que el tercer paso es el que utilizamos para corregir. Destacamos también que la dificultad de este algoritmo de decodificación es encontrar el par corrector de errores (A, B).

La complejidad de este método de decodificación es $\mathcal{O}\left(n^3\right)$, pues el coste de este algoritmo consiste en obtener el núcleo $K_{\mathbf{y}}$, es decir la complejidad de la eliminación gaussiana de una matriz $n' \times n$ con $n' < n$.

3.1.2. Propiedades interesantes para ECP

Veamos algunas propiedades interesantes que pueden facilitarnos la búsqueda de pares correctores de errores. Tendremos en cuenta la siguiente equivalencia:

$$\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp} \Longleftrightarrow (\mathrm{A} * \mathrm{B}) \perp \mathscr{C}.$$

donde $\mathrm{A} \perp \mathrm{B}$ si $\langle \mathbf{a}, \mathbf{b} \rangle = 0$, $\forall \mathbf{a} \in \mathrm{A}, \mathbf{b} \in \mathrm{B}$.

Observación 3.8. Sea $I = \{i_1, \ldots, i_t\}$ con $1 \leq i_1 < \ldots < i_t \leq n$ definimos la proyección $\pi_I \colon \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^t$ por $\pi_I(x) = (x_{i_1}, x_{i_2}, \ldots, x_{i_t})$. Denotaremos $\operatorname{Im} \pi_I = \mathrm{A}_I$ y $\ker \pi_I = \mathrm{A}(I) = \{a \in \mathrm{A} \mid a_i = 0, \forall i \in I\}$. Se demuestra que $\dim(\mathrm{A}_I) = \sharp I$, $\forall I$ con al menos t elementos, si, y sólo si, $\mathrm{d}_{min}(\mathrm{A}^{\perp}) > t$.

Proposición 3.9. *Si* A, B *y* $\mathscr{C}$ *son* $[n,k]_q$ *códigos tales que* $(\mathrm{A} * \mathrm{B}) \perp \mathscr{C}$, $\mathrm{d}_{min}(\mathrm{A}^{\perp}) > a > 0$ *y* $\mathrm{d}_{min}(\mathrm{B})^{\perp} > b > 0$, *entonces,* $\mathrm{d}_{min}(\mathscr{C}) \geq a + b$.

Demostración. Sea $\mathbf{c} \in \mathscr{C} \setminus \{0\}$ con $\operatorname{supp}(\mathbf{c}) = \mathrm{I}$. Consideremos $t = \sharp I$. Sin perdida de generalidad asumimos que $a \leq b$. Entonces,

$$\dim(\mathrm{A}_I) + \dim(\mathrm{B}_I) = \begin{cases} 2t, \text{ si} & t \leq a \\ a + t, \text{ si} & a < t \leq b \\ a + b, \text{ si} & b < t \end{cases}$$

por la Observación 3.8. Pero, $(\mathrm{A} * \mathrm{B}) \perp \mathrm{C}$ por tanto, $(\mathbf{c} * \mathrm{A})_I \perp \mathrm{B}_I$. Además, $\dim((\mathbf{c} * \mathrm{A})_I) = \dim(\mathrm{A}_I)$, ya que $c_i \neq 0, \forall i \in I$. Por esto, obtenemos que $\dim(\mathrm{A}_I) + \dim(\mathrm{B}_I) \leq \sharp I = t$. Esto únicamente es posible en el caso $t \geq a + b$. Luego deducimos que, $\mathrm{d}_{min}(\mathscr{C}) \geq a + b$. □

Proposición 3.10. *Sea* t *un entero positivo. Si* A *es un* $[n, t+1]$ *código con* $\mathrm{d}_{min}(\mathrm{A}) = n - t$ *y* B *es un* $[n,t]$ *código con* $\mathrm{d}_{min}(\mathrm{B}) = n - t + 1$ *ambos sobre* $\mathbb{F}_{q^N}$ *y* $\mathscr{C}$ *es un* $[n,k]_q$ *código tal que* $(\mathrm{A} * \mathrm{B}) \perp \mathscr{C}$ *entonces,* $\mathrm{d}_{min}(\mathscr{C}) \geq 2t + 1$ *y* (A, B) *es un par* t*-corrector de errores para* $\mathscr{C}$ *sobre* $\mathbb{F}_{q^N}$.

Demostración. El código A es un código MDS, por tanto, su dual también es un código MDS de parámetros $[n, n-t-1]$ (Proposición 1.21) y $\mathrm{d}_{min}(\mathrm{A}^{\perp}) = t + 2 > t + 1$.

De la misma forma, obtenemos que $\mathrm{d}_{min}(\mathrm{B}^{\perp}) = t+1 > t$. Por la Proposición 3.9 obtenemos que $\mathrm{d}_{min}(\mathscr{C}) \geq 2t+1$. Es claro que $k(\mathrm{A}) = t+1 > t$. Además, $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) \geq n+t+1 > n$. Concluimos entonces que (A,B) es un par t-corrector de errores para $\mathscr{C}$ sobre $\mathbb{F}_{q^N}$. □

Proposición 3.11. *Sean $\mathscr{C}$ un código $[n, n-2t]$ con $\mathrm{d}_{min}(\mathscr{C}) = 2t+1$ y (A,B) un par t-corrector de errores para $\mathscr{C}$. Entonces,* A *es un código $[n, t+1]$ con $\mathrm{d}_{min}(\mathrm{A}) = n-t$.*

Demostración. Dado que $(\mathrm{A} * \mathrm{B}) \perp \mathscr{C}$ entonces $(\mathrm{B} * \mathscr{C}) \perp \mathrm{A}$. Además, $\mathscr{C}$ es un código MDS, por tanto, $\mathscr{C}^{\perp}$ es también un código MDS de parámetros $[n, 2t]$ (Proposición 1.21) y $\mathrm{d}_{min}(\mathscr{C}^{\perp}) = n-2t+1 > n-2t$.

Como (A,B) es un par t-corrector de errores, sabemos que $\mathrm{d}_{min}(\mathrm{B}^{\perp}) \geq t+1$. Por la Proposición 3.9, obtenemos que $\mathrm{d}_{min}(\mathrm{A}) \geq t + (n-2t) = n-t$. Además, $k(\mathrm{A}) \geq t+1$. Concluimos, por tanto, que A es un código MDS de parámetros $[n, t+1]$ y $\mathrm{d}_{min}(\mathrm{A}) = n-t$. □

Observación 3.12. La condición (4) de la definición 3.3 implica que la aplicación π_I es un isomorfismo entre A y A_I para todo $I = \mathrm{supp}(\mathbf{c})$ con $\mathbf{c} \in \mathscr{C} \setminus \{0\}$. En efecto, sea $\mathbf{c} \in \mathscr{C}$ con $I = \{i \mid c_i \neq 0\}$ y sea $\mathbf{a} \in \mathrm{A}(I) = \ker \pi_I$, es decir $a_i = 0$, $\forall i \in I$. Entonces:

$$n \geq \sharp I + \mathrm{w}_H(a) \geq \mathrm{d}_{min}(\mathscr{C}) + \mathrm{d}_{min}(\mathrm{A})$$

contradiciendo la condición (4).

Proposición 3.13. *Sea $\mathscr{C}$ un $[n,k]$ código con $\mathrm{d}_{min}(\mathscr{C}) = 2t+1$. Si (A,B) es un par t-corrector de errores para $\mathscr{C}$ y $\mathrm{d}_{min}(\mathrm{B}) + \mathrm{d}_{min}(\mathscr{C}) > n$. Entonces,* B *es un código $[n,t]$ con $\mathrm{d}_{min}(\mathrm{B}) = n-t+1$.*

Demostración. Sea $\mathbf{c} \in \mathscr{C} \setminus \{0\}$ de peso mínimo y con $\mathrm{supp}(\mathbf{c}) = I$. Entonces, $\sharp I = 2t+1 = \mathrm{d}_{min}(\mathscr{C})$. Además, $\dim(\mathrm{A}) = \dim(\mathrm{A}_I)$ por la Observación 3.12 y, por la hipótesis $\mathrm{d}_{min}(\mathrm{B}) + \mathrm{d}_{min}(\mathscr{C}) > n$, sabemos que $\dim(\mathrm{B}) = \dim(\mathrm{B}_I)$. Además, como $(\mathrm{A} * \mathrm{B}) \perp \mathscr{C}$ sabemos que $(\mathbf{c} * \mathrm{A})_I \perp \mathrm{B}_I$ en $\mathbb{F}_q^{2t+1}$ por tanto, $\dim(\mathrm{A}_I) + \dim(\mathrm{B}_I) \leq 2t+1$. Entonces,

$$(t+1) + \dim(\mathrm{B}) \leq \dim(\mathrm{A}) + \dim(\mathrm{B}) \leq 2t+1.$$

De este modo, $k(\mathrm{B}) \leq t$. Por tanto, $k(\mathrm{B}^{\perp}) \geq n-t$ y $\mathrm{d}_{min}(\mathrm{B}^{\perp}) \geq t+1$ por la definición de par corrector de errores. Concluimos que $\mathrm{B}^{\perp}$ es un código MDS y, por tanto, por la Proposición 1.21 B también es un código MDS de parámetros $[n,t]$ y $\mathrm{d}_{min}(\mathrm{B}) = n-t+1$. □

3.1.3. Funciones localizadoras de errores.

Definición 3.14. *Sean* A, B *y* $\mathscr{C}$ *códigos lineales de longitud* n *sobre* $\mathbb{F}_q$*. Diremos que* (A, B) *es un par* t*-localizador de errores para* $\mathscr{C}$ *si cumple las siguientes condiciones:*

(E1) $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$
(E2) $k(\mathrm{A}) > t$
(E3) $\mathrm{d}_{min}(\mathrm{B}) > t$

Teorema 3.15. *Sea* (A, B) *un par* t*-localizador de errores para el código* $\mathscr{C}$*. Sea* $\mathbf{y} = \mathbf{c} + \mathbf{e}$ *un vector en* $\mathbb{F}_q^n$ *con* $\mathbf{c} \in \mathscr{C}$ *y* $\mathbf{e} \in \mathbb{F}_q^n$ *con* $\mathrm{w}_H(\mathbf{e}) < t$*. Entonces existe un vector* $\mathbf{a} \in \mathrm{A} \setminus \{0\}$ *tal que:*

$$\sum_{i=0}^{n-1} y_i a_i b_i = 0, \ \forall \mathbf{b} \in \mathrm{B}. \tag{3.1}$$

Además, cada solución $\mathbf{a} \in \mathrm{A}$ *de* (3.1) *satisface:*

$$\mathbf{e} * \mathbf{a} = 0. \tag{3.2}$$

Demostración. La demostración es consecuencia de los Lemas 3.4-3.6

Algoritmo para localizar errores.

Sea (A, B) un par t-localizador de errores para $\mathscr{C}$. Supongamos que recibimos el vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ con $\mathbf{c} \in \mathscr{C}$ y donde $\mathrm{w}_H(\mathbf{e}) < t$. Entonces:

1. Encontrar $\mathbf{a} \in \mathrm{A}$ tal que $\langle \mathbf{e}, \mathbf{a} * \mathbf{b} \rangle = \langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = 0$, $\forall \mathbf{b} \in \mathrm{B}$.
2. Como la condición (3.1) es equivalente a la condición (3.2), entonces, $\mathbf{e} * \mathbf{a} = 0$ y, por tanto, podemos localizar errores.

Expresión matricial del algoritmo que nos permite localizar el error

En primer lugar, consideramos la matriz

$$\mathrm{diag}(\mathbf{y}) = \begin{pmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_n \end{pmatrix} \in \mathbb{F}_q^{n \times n}.$$

Con esto la expresión $\langle \mathbf{y}, \mathbf{a} * \mathbf{b} \rangle = 0$, $\forall \mathbf{b} \in \mathrm{B}$, es equivalente a

$$\mathbf{b} \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{a}^{\perp} = 0, \ \forall \mathbf{b} \in \mathrm{B}.$$

Si $\mathcal{G}_{\mathrm{A}}$, $\mathcal{G}_{\mathrm{B}}$ representan unas matrices generatrices de A y B, respectivamente, y $\sigma \in \mathbb{F}_q^{k(\mathrm{A})}$ un mensaje; podemos reescribir la ecuación anterior como:

$$\mathcal{G}_{\mathrm{B}} \cdot \mathrm{diag}(\mathbf{y}) \cdot \mathbf{a}^{\perp} = \mathcal{G}_{\mathrm{B}} \cdot \mathrm{diag}(\mathbf{y}) \cdot (\sigma\mathcal{G}_{\mathrm{A}})^{\perp} = 0.$$

Por tanto, si llamamos $S(\mathbf{y}) = \mathcal{G}_{\mathrm{B}}\mathrm{diag}(\mathbf{y})\mathcal{G}_{\mathrm{B}}^{\perp}$ nos queda la ecuación:

$$S(\mathbf{y})\sigma^{\perp} = 0. \tag{3.3}$$

Concluimos que, cualquier solución de (3.3) será una palabra $\mathbf{a} = \sigma\mathcal{G}_{\mathrm{A}}$ que nos permitirá localizar errores.

El problema, por tanto, es encontrar (A, B) que cumpla las condiciones de la definición de par t-localizador de errores. La elección de $\mathcal{G}_{\mathrm{A}}$ y $\mathcal{G}_{\mathrm{B}}$ es libre, sin embargo, afectará al coste computacional. Es decir, una elección de matrices en forma sistemática reducirá el número de variables.

Ejemplo 3.16. Consideremos el cuerpo $\mathbb{F}_7$, sea $\mathscr{C}$ el $[7, 3, 5]_7$ código con matriz de paridad,

$$\mathcal{H}_{\mathscr{C}} = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1 \\ 0\,1\,2\,3\,4\,5\,6 \\ 0\,1\,4\,2\,2\,4\,1 \\ 0\,1\,1\,6\,1\,6\,6 \end{pmatrix} \in \mathbb{F}_7^{4\times 7}.$$

Sean A, B los códigos generados por las matrices generatrices

$$\mathcal{G}_{\mathrm{A}} = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1 \\ 0\,1\,2\,3\,4\,5\,6 \\ 0\,1\,4\,2\,2\,4\,1 \end{pmatrix} \in \mathbb{F}_7^{3\times 7} \quad \text{y} \quad \mathcal{G}_{\mathrm{B}} = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1 \\ 0\,1\,2\,3\,4\,5\,6 \end{pmatrix} \in \mathbb{F}_7^{2\times 7}.$$

Es claro que A, B verifican las condiciones (E1)-(E3) para $t \leq 2$. Por lo tanto (A, B) es un par t-localizador de errores.

Sea $\mathbf{y} = \mathbf{c} + \mathbf{e}$ el vector recibido, con $\mathbf{c} \in \mathscr{C}$ y $\mathbf{e} = (0, 3, 0, 0, 1, 0, 0)$. La relación entre el síndrome del vector recibido y la palabra enviada es:

$$S(\mathbf{y}) = S(\mathbf{c}) + S(\mathbf{e})$$

Además la condición (E1) garantiza que $S(\mathbf{c}) = 0$. Calculamos

$$S(\mathbf{y}) = \begin{pmatrix} 4\,0\,5 \\ 0\,5\,4 \end{pmatrix}$$

Una solución σ que cumpla $S(\mathbf{y})\sigma^{\perp} = 0$ es $\sigma = (4, 2, 1)$ que se corresponde con el vector localizador de errores $\mathbf{a} = \sigma\mathcal{G}_{\mathrm{A}} = (4, 0, 5, 5, 0, 4, 3)$.

Observación 3.17. Dado un error particular es fácil probar por el Teorema 3.15 que las siguientes condiciones son suficientes para obtener $\mathbf{a} \in \mathrm{A} \setminus \{0\}$ tal que si se cumple la propiedad (3.2) se verifiquen las siguientes condiciones:

(R1) $\mathscr{C} * \mathrm{A} \subseteq \mathrm{B}^{\perp}$

(R2) $\mathbf{a} \in \mathrm{A} \setminus \{0\}\colon \mathbf{e} * \mathbf{a} = 0$
(R3) $\forall \mathbf{a} \in \mathrm{A} \setminus \{0\}\colon \mathbf{e} * \mathbf{a} \in \mathrm{B}^{\perp}$ entonces $\mathbf{e} * \mathbf{a} = 0$

Las condiciones (R2) y (R3) son más débiles que las condiciones (E2) y (E3), respectivamente, pues se establecen para un cierto modelo de error $\mathbf{e}$.

A continuación, vamos a reformular otras tres condiciones. Éstas aunque van a depender de las posiciones de error no van a depender de sus valores particulares.

Lema 3.18. *Sea* $\mathbf{e} \in \mathbb{F}_q^n$ *un error y sean* E *y* $\bar{E}$ *los conjuntos definidos como*

$$E = \{\mathbf{e}' \in \mathbb{F}_q^n \mid e'_i = 0, \forall i\colon e_i \neq 0 \quad y \quad \bar{E} = \{\mathbf{e}' \in \mathbb{F}_q^n \mid e'_i = 0, \forall i\colon e_i = 0\}$$

entonces las siguientes condiciones son suficientes para localizar las posiciones de error:

(S1) $\mathscr{C} * \mathrm{A} \subseteq \mathrm{B}^{\perp}$
(S2) $\mathrm{A} \cap E \neq 0$
(S3) $\mathrm{B}^{\perp} \cap \bar{E} = 0$

Demostración. Estas condiciones son más débiles que las anteriores (R1)-(R3).

- La primera condición es equivalente a (R1).
- Consideramos $\mathrm{A} \cap E \neq 0$. Esto es, $\exists \mathbf{a} \in \mathrm{A} \setminus \{0\}\colon \mathbf{a} \in \mathrm{A} \cap E$. Entonces, $\mathbf{a} \in E$, es decir, $a_i = 0 \forall i\colon e_i \neq 0$. Por tanto, $\mathbf{a} * \mathbf{e}$, luego (S2) implica (R2).
- Veamos que $\forall \mathbf{a} \in \mathrm{A} \setminus \{0\}\colon \mathbf{e} * \mathbf{a} \in \mathrm{B}^{\perp}$ entonces $\mathbf{e} * \mathbf{a}$. Consideramos los elementos de la forma $\mathbf{a} \in \mathrm{A} \setminus \{0\}\colon \mathbf{e} * \mathbf{a} \in \mathrm{B}^{\perp}$. Como $\mathrm{B}^{\perp} \cap \bar{E}$ entonces $\mathbf{e} * \mathbf{a} = \mathbf{d}$ con $d_i = 0, \forall i, : e_i = 0$, es decir, $\mathbf{e} * \mathbf{a} = \mathbf{d} \in \bar{E}$. Concluyendo que $\mathbf{e} * \mathbf{a} = 0$. Luego (S3) implica (R3).

□

Las condiciones de este lema se pueden expresar en términos de funciones tal y cómo veremos en las siguientes líneas. Consideramos $(S, *) \subseteq \mathbb{F}_q^n$ un subconjunto de $\mathbb{F}_q^n$ con la operación $*$. Denotaremos por R al anillo de funciones tal que la aplicación evaluación $\mathrm{Ev} : R \longrightarrow S$ que evalúa la función $f \in R$ en un conjunto de puntos es un homomorfismo sobreyectivo con $\ker(\mathrm{Ev}) = I$.

Definición 3.19 (Funciones asociadas a un código). *Sea* $\mathscr{C} \subset S$ *un código definimos* $L(\mathscr{C}) \subset R$ *como el conjunto de funciones tales que* $\mathrm{Ev}_{|L(\mathscr{C})} : L(\mathscr{C}) \longrightarrow \mathscr{C}$ *es un isomorfismo. En este caso,* $L(\mathscr{C}) \cap I = \langle 0 \rangle$ *pues* $\mathrm{Ev}_{|L(\mathscr{C})}$ *es inyectivo.*

Ejemplo 3.20. En el caso de códigos cíclicos. Sean $R = \mathbb{F}_q[x]$ y $\alpha \in \mathbb{F}_q$ una raíz primitiva de la unidad. Consideramos

$$\begin{aligned} \mathrm{Ev}\colon \mathbb{F}_q[x] &\longrightarrow \mathbb{F}_q^n \\ f &\longmapsto (f(1), f(\alpha), \ldots, f(\alpha^n - 1)) \end{aligned}$$

con $\ker \mathrm{Ev} = \{f \in \mathbb{F}_q[x]\colon \mathrm{Ev}(f) = 0\} = \langle x^n - 1 \rangle$.

Por lo tanto, $\mathrm{Ev}\colon \frac{\mathbb{F}_q[x]}{\langle x^n - 1 \rangle} \longrightarrow \mathrm{Im}\,(\mathrm{Ev}) \subseteq \mathbb{F}_q^n$ es un isomorfismo.

Lema 3.21. *Sean* $L(\mathrm{A}), L(\mathrm{B})$ *y* $L(\mathscr{C})$ *conjunto de funciones asociadas a los códigos* A, B *y* $\mathscr{C}$ *siguiendo la Definición 3.19. Definimos* $\forall \mathbf{e} \in \mathbb{F}_q^n$ *los siguientes ideales de* R*:*

$$\begin{cases} J = \langle \{ f \in R \mid (\mathrm{Ev}(f)_i) = 0, \forall i \colon e_i \neq 0 \} \rangle \\ \bar{J} = \langle \{ f \in R \mid (\mathrm{Ev}(f)_i) = 0, \forall i \colon e_i = 0 \} \rangle \end{cases}.$$

Las siguientes condiciones son suficientes para localizar las posiciones de error:

(T1) $L(\mathscr{C}) * L(\mathrm{A}) \subseteq L(\mathrm{B}^{\perp}) + I$
(T2) $L(\mathrm{A}) \cap J \neq \langle 0 \rangle$
(T3) $L(\mathrm{B}^{\perp}) \cap \bar{J} = \langle 0 \rangle$

Demostración. Inmediata por el Lema 3.18. □

Funciones correctoras de errores.

Lema 3.22. *Sean* (A, B) *un par* t*-localizador de errores para* $\mathscr{C}$ *y* $\mathbf{a} \in \mathrm{A} \setminus \{0\} \colon \mathbf{e} * \mathbf{a}$ *con* $\mathbf{e} \in \mathbb{F}_q^n$*, es decir el soporte de* $\mathbf{a}$ *nos permite localizar errores.*

Entonces, los valores del error se pueden determinar de forma única si, y sólo si, $\forall \mathbf{c} \in \mathscr{C}$ *tal que* $\mathbf{c} * \mathbf{a}$ *se tiene que* $\mathbf{c} = 0$*.*

Demostración. Supongamos que podemos escribir $\mathbf{y} = \mathbf{e_1} + \mathbf{c_1} = \mathbf{e_2} + \mathbf{c_2}$ con $\mathbf{c_1}, \mathbf{c_2} \in \mathscr{C}$ y $\mathbf{e_1} * \mathbf{a} = \mathbf{e_2} * \mathbf{a} = 0$. Entonces, $(\mathbf{e_1} - \mathbf{e_2}) * \mathbf{a} = 0$ con $\mathbf{e_1} - \mathbf{e_2} = \mathbf{c_1} - \mathbf{c_2} \in \mathscr{C}$. Si además se verifica que $\forall \mathbf{c} \in \mathscr{C}$ tal que $\mathbf{c} * \mathbf{a}$ se tiene que $\mathbf{e_1} - \mathbf{e_2} = 0$ es decir, $\mathbf{e_1} = \mathbf{e_2}$. En caso contrario, $\exists \mathbf{c} \in \mathscr{C} \colon \mathbf{c} * \mathbf{a} = 0$ con $\mathbf{c} \neq 0$. Por tanto, $\mathbf{e}$ es una solución y $\mathbf{e} - \mathbf{c}$ es otra solución pues $(\mathbf{e} - \mathbf{c}) * \mathbf{a} = 0$. □

Definición 3.23. *Sea* (A, B) *un par* t*-localizador que cumple las condiciones de la Definición 3.14. Diremos que* (A, B) *es un par* t*-corrector de errores si además de las condiciones de la definición cumple que*

$$\mathrm{d}_{min}(\mathscr{C}) + \mathrm{d}_{min}(\mathrm{A}) > n \tag{3.4}$$

donde n *denota la longitud de* $\mathscr{C}$*.*

Esta definición está justificada por el Lema 3.22 puesto que la condición (3.4) implica que

$$\forall \mathbf{c} \in \mathscr{C}, \forall \mathbf{a} \in \mathrm{A} \colon \mathbf{c} * \mathbf{a} = 0 \text{ se tiene que } \mathbf{c} = 0 \tag{3.5}$$

Observación 3.24. El Lema 3.22 implica que para que (A, B) sea un par t-corrector de errores se debe verificar:

$$(\mathscr{C} \setminus \{0\}) * (\mathrm{A} \setminus \{0\}) \subseteq (\mathrm{B}^{\perp} \setminus \{0\})$$

Que en términos de funciones se traduce en,

$$L(\mathscr{C}) * L(\mathrm{A}) \subseteq L(\mathrm{B}^{\perp})$$

donde hemos utilizado que R no tenga divisores de cero.

Utilizando las condiciones (T1)-(T3) se tiene que las siguientes condiciones son suficientes para que (A, B) sea un par t-localizador de errores:

(F1) $L(\mathscr{C}) * L(\mathrm{A}) \subseteq L(\mathrm{B}^{\perp})$
(F2) $L(\mathrm{A}) \cap J \neq \langle 0 \rangle$
(F3) $\langle L(\mathscr{C}) * L(\mathrm{A}) \rangle \cap \bar{J} = \langle 0 \rangle$

Por tanto, el dilema es claro, (F2) implica que $L(\mathrm{A})$ debe tener un gran tamaño mientras que (F3) nos sugiere que $L(\mathrm{A})$ debe ser pequeño.

3.2. Pares correctores de errores para códigos GRS

En esta sección estudiaremos la existencia de ECP para códigos GRS. Recordemos que la definición de estos códigos ya se estudió en el capítulo 2, Definición 2.61.

Proposición 3.25. *[1, Proposition 5.1.26] Sea* $\mathbf{b}^{\perp}$ *el vector con coordenadas:*

$$b_j^{\perp} = \frac{1}{b_j \prod_{i \neq j}(a_i - a_j)}, \forall j = 1, \ldots, n.$$

Entonces, $(\boldsymbol{GRS}_k(\mathbf{a}, \mathbf{b}))^{\perp} = \boldsymbol{GRS}_{n-k}(\mathbf{a}, \mathbf{b}^{\perp})$

Proposición 3.26. $\boldsymbol{GRS}_k(\mathbf{a}, \mathbf{b}) * \boldsymbol{GRS}_l(\mathbf{a}, \mathbf{c}) = \boldsymbol{GRS}_{k+l-1}(\mathbf{a}, \mathbf{b} * \mathbf{c})$

Demostración. Por un lado, $\forall \mathbf{c}_1 \in \mathbf{GRS}_k(\mathbf{a}, \mathbf{b})$ y $\forall \mathbf{c}_2 \in \mathbf{GRS}_l(\mathbf{a}, \mathbf{c})$ se tiene que:

$$\begin{cases} \mathbf{c}_1 = \mathrm{ev}_{\mathbf{a},\mathbf{b}}(f(x)) = \mathbf{b} * f(\mathbf{a}) \text{ con } \deg(f) < k \\ \mathbf{c}_2 = \mathrm{ev}_{\mathbf{a},\mathbf{c}}(g(x)) = \mathbf{c} * g(\mathbf{a}), \text{ con } \deg(g) < l. \end{cases}$$

Por tanto:

$$\mathbf{c}_1 * \mathbf{c}_2 = \mathrm{ev}_{\mathbf{a},\mathbf{b}}(f(x)) * \mathrm{ev}_{\mathbf{a},\mathbf{c}}(g(x)) = (\mathbf{b} * \mathbf{c}) * (fg)(\mathbf{a}), \text{ con } \deg(fg) < k + l - 1$$

Por otro lado, $\mathrm{ev}_{\mathbf{a},\mathbf{b}*\mathbf{c}}(x^{i+j}) = \mathrm{ev}_{\mathbf{a},\mathbf{b}}(x^i) * \mathrm{ev}_{\mathbf{a},\mathbf{c}}(x^j)$ con $1 \leq i < k$, $1 \leq j < l$. □

Sabiendo esto, consideraremos el siguiente par corrector de errores para $\mathbf{GRS}_k(\mathbf{a}, \mathbf{b})$ que nos permitirá decodificar hasta $\mathrm{t} < \left\lfloor \frac{\mathrm{d}_{min}(\mathscr{C})-1}{2} \right\rfloor$.

Proposición 3.27. *Consideramos* $\mathscr{C} = \boldsymbol{GRS}_k(\mathbf{c}, \mathbf{d})$ *con* $\mathrm{d}_{min}(\mathscr{C}) = n - k + 1$. *Sea* $t = \left\lfloor \frac{\mathrm{d}_{min}(\mathscr{C})-1}{2} \right\rfloor$ *y* $\mathscr{C}^{\perp} = \boldsymbol{GRS}_{n-k}(\mathbf{c}, \mathbf{d}^{\perp})$. *Sean*

$$\mathrm{A} = \boldsymbol{GRS}_{t+1}(\mathbf{c}, \mathbf{d}^{\perp}) \quad y \quad \mathrm{B} = \boldsymbol{GRS}_t(\mathbf{c}, \mathbf{1})$$

entonces, (A, B) *es un par* t*-corrector de errores para* $\mathscr{C}$.

Demostración. Veamos que el par (A, B) cumple las condiciones de la Definición 3.3. Tengamos en cuenta que, $2t = \mathrm{d}_{min}(\mathscr{C}) - 1 = n - k$. Por tanto, $\mathscr{C}^{\perp} = \mathbf{GRS}_{2t}(c, d^{\perp})$.

(1) Veamos que $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$. En efecto, pues por la Prop. 3.26 se tiene que:

$$\mathbf{GRS}_{t+1}(\mathbf{c}, \mathbf{d}^{\perp}) * \mathbf{GRS}_{t}(\mathbf{c}, \mathbf{1}) = \mathbf{GRS}_{t+1+t-1}(\mathbf{c}, \mathbf{d}^{\perp} * \mathbf{1}) = \mathbf{GRS}_{2t}(\mathbf{c}, \mathbf{d}^{\perp}) = \mathscr{C}^{\perp}$$

(2) Veamos que $k(\mathrm{A}) > t$. Como $\mathrm{A} = \mathbf{GRS}_{t+1}(\mathbf{c}, \mathbf{d}^{\perp})$, se tiene que $k(\mathrm{A}) = t + 1 > t$.

(3) Veamos que $\mathrm{d}_{min}(\mathrm{B}^{\perp}) > t$. Como $\mathrm{B} = \mathbf{GRS}_{t}(\mathbf{c}, \mathbf{1})$, entonces $\mathrm{B}^{\perp} = \mathbf{GRS}_{n-t}(c, 1)$ es un código MDS con $k = n - t$ por la Proposición 2.57. Por tanto, $\mathrm{d}_{min}(\mathrm{B}^{\perp}) = n - (n - t) + 1 = t + 1 > t$.

(4) Faltaría comprobar que $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) > n$. En efecto, sabemos que A y $\mathscr{C}$ son códigos MDS por la Proposición 2.57. Entonces:

$$\mathrm{d}_{min}(\mathrm{A}) = n - (t + 1) + 1 = n - t \quad \text{y} \quad \mathrm{d}_{min}(\mathscr{C}) = n - k + 1 = 2t + 1$$

Por tanto, $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) = n - t + 2t + 1 = n + t + 1 > n$.

□

3.3. Pares correctores de errores para códigos BCH

A lo largo de esta sección trabajaremos con códigos cíclicos $\mathscr{C} \subset \mathbb{F}_q^n$. Recordemos que un código cíclico puede verse como un ideal en $\frac{\mathbb{F}_q[x]}{\langle x^n - 1 \rangle}$ y que podemos hablar de polinomio generador $\mathscr{C} = \langle g(x) \rangle$.

Supondremos que $\mathrm{mcd}(\mathrm{car}(\mathbb{F}_q), n) = 1$ por tanto, $x^n - 1$ tiene n ceros distintos. Consideremos $\bar{\mathbb{F}}_q$ la extensión de $\mathbb{F}_q$. Tomamos $\alpha \in \bar{\mathbb{F}}$ raíz primitiva n-ésima de la unidad y definimos por $m_{\alpha^i}(x)$ el polinomio mínimo de α^i sobre $\mathbb{F}_q$.

Definición 3.28 (Conjunto de control de $\mathscr{C}$). *Si*

$$g(x) = \mathrm{mcm}\{m_{\alpha^i}(x) \colon i \in \mathcal{Z}(\mathscr{C})\},$$

entonces diremos que $\mathcal{Z}(\mathscr{C}) = P$ es un conjunto de índices de control *para el código $\mathscr{C}$. Es decir, el conjunto $P = \{i_1, \cdots, i_l\}$ proporcionará la siguiente matriz de paridad de $\hat{\mathscr{C}} \subset \bar{\mathbb{F}}^n$ (donde $\mathscr{C} = \hat{\mathscr{C}} \cap \mathbb{F}_q^n$),*

$$\mathcal{H}(P) = \begin{pmatrix} (\alpha^{i_1})^0 & (\alpha^{i_1})^1 & \cdots & (\alpha^{i_1})^n \\ (\alpha^{i_2})^0 & (\alpha^{i_2})^1 & \cdots & (\alpha^{i_2})^n \\ \vdots & \vdots & \ddots & \vdots \\ (\alpha^{i_n})^0 & (\alpha^{i_n})^1 & \cdots & (\alpha^{i_n})^n \end{pmatrix}$$

Definición 3.29 (Conjunto generador de $\mathscr{C}$). *Si utilizamos la matriz $\mathcal{H}(P)$ como una matriz generatriz del código $\mathscr{C}$, entonces diremos que P es un conjunto de índices generador del código $\mathscr{C}$.*

En otras palabras dado un conjunto de índices P, diremos que P es generador del código $\mathscr{C}$ si $\mathcal{H}(P)$ es una matriz generatriz de dicho código. En este caso, P será también un conjunto de control del código $\mathscr{C}^{\perp}$.

Definición 3.30. *Sea I un conjunto generador de un código cíclico* A *definimos*

$$\mathrm{A} = \{\mathbf{a} \in \bar{\mathbb{F}}_q^n : \mathbf{a} = \sigma\mathcal{H}(I), \sigma \in \bar{\mathbb{F}}_q^{\sharp I}\}.$$

Es fácil comprobar que si I es un conjunto generador de A entonces, $k(\mathrm{A}) = \sharp I$. Por otro lado, sea $I = \{i_1, \cdots, i_l\}$ con $i_1 < \ldots < i_l$ definimos $\bar{I}$ como el menor conjunto que contiene a los enteros positivos de I, es decir,

$$\bar{I} = \{i_1, i_1 + 1, \cdots, i_l - 1, i_l\}.$$

Sabiendo esto, podemos reformular la cota BCH.

Lema 3.31. *La distancia mínima de un código cíclico de longitud n con conjunto generador I esta acotada por,*

$$\mathrm{d}_{min}(\mathrm{A}) \geq n - \sharp\,\bar{I} + 1.$$

Demostración. Consecuencia de la Cota BCH, Teorema 2.52. □

Observación 3.32. Es fácil comprobar que si $\mathrm{A}, \mathrm{B}, \mathrm{C}$ son códigos cíclicos con conjuntos generadores $I, J, I + J$, respectivamente. Entonces, $\mathrm{A} * \mathrm{B} \subseteq \mathrm{C}$.

Lema 3.33 (Cota de Roos). *[12, Theorem3] Sea I un conjunto de control para un código cíclico* A *con distancia mínima $d_{min}(\mathrm{A})$. Si el conjunto J satisface*

$$\sharp\,\bar{J} \leq \sharp\,J + \mathrm{d}_{min}(\mathrm{A}) - 2 \tag{3.6}$$

entonces, el código con conjunto de control $A + B$ satisface que

$$\mathrm{d}_{min}(\mathrm{A} + \mathrm{B}) \geq \sharp\,J + \mathrm{d}_{min}(A) - 1$$

Teorema 3.34. *Sean $s < t$ y I, J, K conjuntos de control tales que*

$$\sharp\,I = t + 1, \sharp\,J = t - s, \sharp\,\bar{J} = t - s, \sharp\,K = s + 1 \quad y \quad \sharp\,\bar{K} \leq t.$$

Entonces el código $\mathscr{C}$ con conjunto de control $P = I + J + K$ tiene un par t-localizador de errores (A, B) *donde* A, B *tienen conjuntos generadores I y $J + K$, respectivamente.*

Para la distancia del código sabemos que si $\sharp\,\bar{I} \leq 2t$ entonces, $\mathrm{d}_{min}(\mathscr{C}) \geq 2t + 1$. Además, el par (A, B) *es t-corrector de errores cuando*

$$\sharp\,\bar{I} \leq \mathrm{d}_{min}(\mathscr{C}).$$

Demostración. Veamos que (A, B) es un par localizador de errores para $\mathscr{C}$.

1. Veamos que $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^{\perp}$. Sabemos que $\mathcal{H}(I+J+K)$ es matriz de paridad para $\mathscr{C}$ y, por tanto, matriz generatriz de $\mathscr{C}^{\perp}$ entonces, $I+J+K$ es conjunto generador de $\mathscr{C}^{\perp}$. Concluimos aplicando la Observación 3.32.
2. Veamos que $k(\mathrm{A}) > t$. Efectivamente, $k(\mathrm{A}) = \sharp\, I = t+1 > t$.
3. Veamos que $\mathrm{d}_{min}(\mathrm{B}^{\perp}) > t$. Consideramos el código D con conjunto de control J. Por la cota de Singleton, Proposición 1.19, se tiene que:

$$\mathrm{d}_{min}(\mathrm{D}) \leq n - (n - \sharp\, J) + 1 = \sharp\, J + 1 = t - s + 1$$

Y por la cota BCH, Teorema 2.52, se tiene que:

$$\mathrm{d}_{min}(\mathrm{D}) \geq \sharp\, \bar{J} + 1 = t - s + 1$$

Por tanto, $\mathrm{d}_{min}(\mathrm{D}) = t - s + 1$. Además el conjunto de control K cumple la condición (3.6) de la cota de Roos, es decir:

$$\sharp\, \bar{K} \leq \sharp\, K + \mathrm{d}_{min}(\mathrm{D}) - 2 = (s+1) + (t-s+1) - 2 = t.$$

Por tanto, aplicando la cota de Roos (Lema 3.33) se tiene que:

$$\mathrm{d}_{min}(\mathrm{B}^{\perp}) = \mathrm{d}_{min}(J+K) \geq \sharp\, K + \mathrm{d}_{min}(\mathrm{D}) - 1 = (s+1) + (t-s+1) - 1 = t+1.$$

Por lo tanto hemos comprobado que el par (A, B) es t-localizador de errores. Veamos que si $\sharp\, \bar{I} \leq 2t$, entonces también es t-corrector de errores.

4. Es decir, faltaría comprobar que $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) > n$. En primer lugar, calculemos $\mathrm{d}_{min}(\mathscr{C})$ aplicando la cota de Roos (Lema 3.33). Tomamos el código D con conjunto de control $J+K$, entonces $\mathrm{d}_{min}(D) \geq (t-s) + (s+1) + 1 = t+2$.
Es fácil comprobar que el conjunto de control I cumple la condición (3.6) de la cota de Roos, esto es, $\sharp\, \bar{I} \leq \sharp\, I + \mathrm{d}_{min}(\mathrm{D}) - 2 = (t+1) + (t+1) - 2 = 2t$. Por tanto,

$$\mathrm{d}_{min}(\mathrm{C}) = \mathrm{d}_{min}(I+J+K) \geq \sharp\, I + \mathrm{d}_{min}(\mathrm{D}) - 1 = (t+1) + (t+1) - 1 = 2t+1.$$

Concluimos utilizando la cota BCH reformulada (Lema 3.31) que

$$\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) = (n - \sharp\, \bar{I} + 1) + (2t+1) = (n-2t+1) + (2t+1) = n+2 > n.$$

□

Corolario 3.35 ([7]). *Consideremos s tal que $\mathrm{d}_{min}(\mathscr{C}) \geq d_0 + s = 2t+1$ y que verifica que $s+1 \leq d_0 - 1$. Definimos los conjuntos generadores I, J, K como,*

$$I = \{0, 1, 2, \cdots, t\},\ J = \{1, 2, \cdots, t-s\},\ K = \{j_1, j_2, \cdots, j_{s+1}\}$$

Entonces el código $\mathscr{C}$ con conjunto de control $P = I + J + K$ tiene distancia $\mathrm{d}_{min}(\mathscr{C}) \geq 2t+1$ y además, (A, B) es un par t-corrector de errores cuando A, B tienen conjuntos generadores I y $J+K$, respectivamente.

Bibliografía

[1] Pellikaan, R., Wu, X., Bulygin, S. y Jurrius, R.(2017)*Codes, Cryptology and Curves with Computer Algebra.* In press.

[2] Huffman, W. y Pless, V.(2003)*Fundamentals of Error-Correcting Codes.* Cambridge: Cambridge Univ. press

[3] Pellikaan, R. (1996) *On the existence of error-correcting pairs.* Journal of Statistical Planning and Inference, 51, 229-242.

[4] Pellikaan, R. (1993) *On the efficient decoding of algebraic-geometric codes.* in Eurocode 92, CISM Courses and Lectures, 339, 231-253.

[5] Kötter, R. (1992) *A unified description of an error locating procedure for linear codes.* Proc. Algebraic and Combinatorial Coding Theory, 113-117.

[6] Duursma, I.M. (1993) *Decoding codes from curves and cyclic codes.* PhD. thesis, Eindhoven University of Technology.

[7] Duursma, I.M. y Kötter, R. (1994) *Error-locating pairs for cyclic codes.* IEEE Trans. Inform. Theory, 40, 1108-1121.

[8] McEliece, R.J (1978) *A public-key cryptosystem based on algebraic coding theory.* DSN Progress Report, 42-44, 114-116.

[9] Berlekamp, E., McEliece, R., van Tilborg, H. (1978) *On the inherent intractability of certain coding problems* IEEE Transactions on Information, 24, 384-386.

[10] Diffie, W., Hellman, M.E. (1982) *New directions in cryptography.* In: Secure communications and asymmetric cryptosystems, AAAS Sel. Sympos. Ser., 69, 143-180.

[11] Justesen J. y Hoholdt T.(2003)*A Course In Error-Correcting Codes.* Switzerland: European Mathematical Society.

[12] van Lint, J.H y Wilson, R.M(1986)*On the minimum distance of cyclic codes* IEEE Trans. Inform. Theory, 40, 23-40.

Algebraic decoding of cyclic codes.

Universidad de La Laguna

Given any cyclic code: it is easy correcting errors?

Cathaysa Pérez Quintero

Facultad de Ciencias · Sección de Matemáticas

Universidad de La Laguna

alu0100824230@ull.edu.es

FACULTAD DE CIENCIAS

Abstract

The main goal of coding theory is to efficiently transfer reliable information between sender and receiver. We need codes with high information rate, the algorithm used should have a high error correction capacity and the complexity of coding and decoding should be low. An example of decoding algorithms are those that use error-correcting pairs (ECP). In this thesis degree we will characterize ECP for cyclic codes, in particular for BCH and generalized Reed-Solomon (GRS) codes.

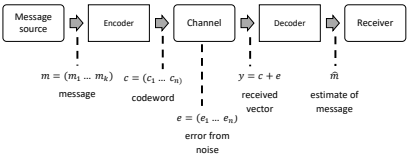

Figure 1: *The alphabet will be a finite set of letters, we use the finite field $\mathbb{F}_q$ with q elements, and the code $\mathscr{C}$ will be a finite set of words. The original message $\mathbf{m}$ will be vectors in the space $\mathbb{F}_q^k$ and the codewords will be vectors in the space $\mathbb{F}_q^n$. The encoder will be a function with $[n,k]_q$ parameters, $\mathrm{Enc}\colon \mathbb{F}_q^k \longrightarrow \mathbb{F}_q^n$ with $k < n$. The codeword is sent over the channel with noise in the form of an error vector $\mathbf{e} \in \mathbb{F}_q^n$, and we will receive a vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in \mathscr{C}$. The decoder will be a function $\mathrm{Dec}\colon \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^k$ that retrieves the original message.*

1. Linear codes

An $[n,k]_q$ **linear code** $\mathscr{C}$ over the finite field $\mathbb{F}_q$ is a subset of $\mathbb{F}_q^n$ of dimention k. A **generator matrix** for an $[n,k]_q$ code $\mathscr{C}$ is any $k \times n$ matrix $\mathscr{G}$ whose rows form a basis for $\mathscr{C}$. A **parity check matrix** for an $[n,k]_q$ code $\mathscr{C}$ is any $(n-k) \times n$ matrix $\mathscr{H}$ such that $\mathscr{G}\mathscr{H}^T = 0$. The **dual code** for an $[n,k]_q$ code $\mathscr{C}$ is $\mathscr{C}^\perp = \{h \in \mathbb{F}^n \mid \langle h, \mathbf{c}\rangle = 0, \forall \mathbf{c} \in \mathscr{C}\}$. The **Hamming distance** between two vectors x and y, denoted $\mathrm{d}_H(x,y)$ is the number of coordinates where they differ. The **minimum distance of a code** $\mathscr{C}$, $\mathrm{d}_{min}(\mathscr{C})$, is the minimum Hamming distance between any pair of different codewords.**Encoding.** We can encode with the function:

$$\begin{aligned} \mathrm{Enc}\colon\ \mathbb{F}_q^k &\longrightarrow \mathbb{F}_q^n \\ \mathbf{m} &\longmapsto \mathbf{m}\mathscr{G} \in \mathscr{C}. \end{aligned}$$

Decoding. Is a NP-complete problem. We can decode with the function:

$$\begin{aligned} \mathrm{Dec}\colon\ \mathbb{F}_q^n &\longrightarrow \mathbb{F}_q^k \\ \mathbf{y} = \mathbf{c} + \mathbf{e} &\longmapsto \mathrm{Dec}(\mathbf{y}) = \mathbf{m}\colon \mathbf{m}\mathscr{G} \in \mathscr{C}. \end{aligned}$$

2. Cyclic codes

Theorem The code $\mathscr{C} \subseteq \mathbb{F}_q[x]_{<n}$ is a cyclic code if and only if $\mathscr{C}$ is an ideal of $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$.

Theorem (Fundamental theorem of cyclic codes) Let $\mathscr{C}$ a cyclic code, exists $g(x) \in \mathscr{C}$ that satisfies the following conditions:

1. $g(x)$ is the only monic polynomial with minimal degree in $\mathscr{C}$.
2. $\mathscr{C} = \langle g(x)\rangle$
3. $g(x)$ divides $x^n - 1$.

This polynomial $g(x)$ is defined as the **generating polynomial** of $\mathscr{C}$. We define the **check polynomial** of $\mathscr{C} = \langle g(x)\rangle$ as

$$h(x) = \frac{x^n-1}{g(x)} = h_0 + h_1 x + \ldots + h_k x^k.$$

We define the **generator matrix** and **parity check matrix** as

$$\mathscr{G} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & g_2 & \cdots & g_{n-k} \end{pmatrix} \text{ y } \mathscr{H} = \begin{pmatrix} h_k & h_{k-1} & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & \cdots & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_k & h_{k-1} & \cdots & h_0 \end{pmatrix}$$

3. Error correcting pairs

Let $\mathscr{C}$ be a linear code in $\mathbb{F}_q^n$ and A, B be linear codes in $\mathbb{F}_{q^m}^n$. Then, (A,B) is a t-ECP for $\mathscr{C}$ if the following conditions hold:

(1) $\mathrm{A} * \mathrm{B} \subseteq \mathscr{C}^\perp$
(2) $k(\mathrm{A}) > t$
(3) $\mathrm{d}_{min}(\mathrm{B}^\perp) > t$
(4) $\mathrm{d}_{min}(\mathrm{A}) + \mathrm{d}_{min}(\mathscr{C}) > n$

4. Error correcting pairs for generalized Reed-Solomon codes

Let $n = q-1$, $1 \le k \le n$ if $\mathbf{a} = (a_1, a_2, \cdots, a_n) \in \mathbb{F}_q^n$ with $a_i \neq a_j$, $\forall i \neq j$ and if $\mathbf{b} = (b_1, b_2, \cdots, b_n) \in \mathbb{F}_q^n$ with $b_i \neq 0$, we define the evaluation map:

$$\begin{aligned} ev_{\mathbf{a},\mathbf{b}}\colon\ \mathbb{F}_q[x] &\longrightarrow \mathbb{F}_q^n \\ f(x) &\longmapsto \mathbf{b} * f(\mathbf{a}) = (b_1 f(a_1), b_2 f(a_2), \ldots, b_n f(a_n)) \end{aligned}$$

And we define GRS codes as

$$GRS_k(\mathbf{a},\mathbf{b}) = \{ev_{\mathbf{a},\mathbf{b}}(f) \mid f(x) \in L_k\}.$$

Proposition. Let $\mathscr{C} = GRS_k(\mathbf{c},\mathbf{d})$ with $\mathrm{d}_{min}(\mathscr{C}) = n-k+1$. Let $t = \left\lfloor \frac{\mathrm{d}_{min}(\mathscr{C})-1}{2} \right\rfloor$ and $\mathscr{C}^\perp = GRS_{n-k}(\mathbf{c},\mathbf{d}^\perp)$. Then

$$\mathrm{A} = GRS_{t+1}(\mathbf{c},\mathbf{d}^\perp) \quad \text{and} \quad \mathrm{B} = GRS_t(\mathbf{c},\mathbf{1})$$

then, (A,B) is a t-error correcting pair for $\mathscr{C}$.

5. Error correcting pairs for BCH codes

We define a **BCH** code with designed minimum distance δ as a cyclic code $\mathscr{C}$ with set of zeros $\mathscr{Z}(\mathscr{C}) \supseteq \{\beta^b, \beta^{b+1}, \cdots, \beta^{b+\delta-2} \mid \beta \in \mathbb{F}_{q^m}\}$.

Proposition Let s such that $\mathrm{d}_{min}(\mathscr{C}) \ge d_0 + s = 2t+1$ and $s+1 \le d_0 - 1$. Let I, J, K generating sets such that,

$$I = \{0,1,2,\cdots,t\},\ J = \{1,2,\cdots,t-s\},\ \text{and}\ K = \{j_1, j_2, \cdots, j_{s+1}\}$$

Then the code $\mathscr{C}$ with check set $P = I + J + K$ have minimun distance $\mathrm{d}_{min}(\mathscr{C}) \ge 2t+1$. Furthermore, let A,B codes with generating sets I and $J+K$, respectively. Then (A,B) is a t-ECP for $\mathscr{C}$.

REFERENCES: DUURSMA, I.M. Y KÖTTER, R. (1994) *Error-locating pairs for cyclic codes.* IEEE Trans. Inform. Theory, 40, 1108-1121.