INTRODUCCIÓN A LA TEORÍA DE CÓDIGOS AUTOCORRECTORES

RICARDO A. PODESTÁ

Resumen. En estas notas se presenta una introducción a la teoría de los códigos autocorrectores a través del estudio de una clase particular muy importante es éstos, los llamados códigos lineales. Se introducen algunos conceptos y resultados básicos de la teoría y se estudian en mayor detalle algunas familias famosas de tales códigos.

Índice

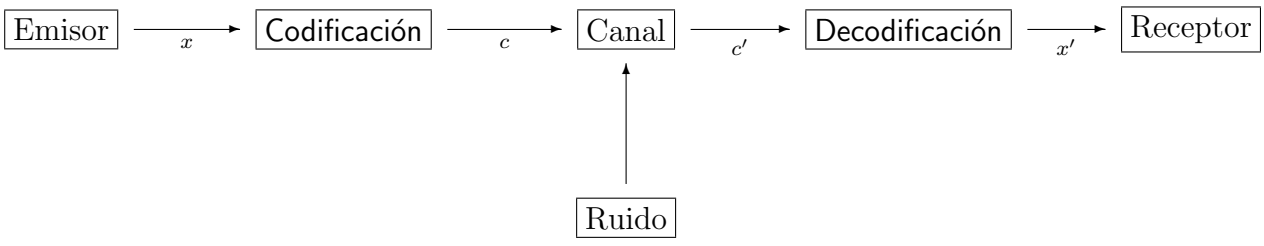
Date: 6 de junio de 2006 (6/6/6).
1991 *Mathematics Subject Classification.* 94-01, 94-06, 94B05, 94B15.
CONICET y SecytUNC.

Introducción

Como el título indica, estas notas tratan sobre *Códigos Autocorrectores*. ¿Contienen alguna revelación sobre el *Código Da Vinci*? ¿Tienen algo que ver con los "códigos del fútbol"? No, en absoluto. Por otra parte, existen ciertos tipos de códigos como los lenguajes naturales, la notación musical y las señales de tránsito, que no son códigos en el sentido matemático que nos interesa. Por ejemplo, los lenguajes naturales son códigos (técnicamente hablando, no-lineales y de longitud variable) aunque con muy malas propiedades en cuanto a detección y corrección de errores. Casi siempre, por sintaxis o contexto, podemos detectar un error, pero es muy difícil corregirlo. Como ejemplo, un hombre va a buscar a su amada a la habitación y solo encuentra una escueta notita que dice "`Xo te amo`". El hombre detecta el error, pero duda entre interpretar el texto como "`Yo te amo`" o interpretarlo como "`No te amo`". En inglés, el ejemplo se torna más dramático, ya que el hombre encuentra la nota "`I love Xou`", con las posibles interpretaciones "`I love You`" o "`I love Lou`" (quien además podría ser su amigo...)

Para despejar estas dudas, en este apunte explicamos a que cosa nos referimos por código y qué quiere decir que éste sea autocorrector. Trataremos también de mostrar para qué sirven y cuál es el interés en ellos. El objetivo del curso que presentamos, es dar una breve introducción a la teoría de los códigos autocorrectores, a través de numerosos ejemplos y de una clase particular muy rica de códigos, los códigos lineales. La escasez de tiempo y espacio han hecho que muchos temas queden sin tratar. Algunos, como los códigos perfectos o los enumeradores de peso y la identidad de MacWilliams son tratados, aunque no en el detalle que estos se merecen. He preferido incluir un poco de información extra, a costa de dejar algunos hechos sin demostración. La frase "ejercicio para el lector" será, por lo tanto, frecuentemente utilizada.

La situación general es, *grosso modo*, la siguiente. Supongamos que queremos enviar un mensaje. Éste es enviado por un *canal de comunicación*, cuyas características dependen de la naturaleza del mensaje a ser enviado (i.e. sonido, imagen, datos). En general, hay que hacer una *traducción* entre el mensaje original (o *palabra fuente*) x y el tipo de mensaje c que el canal está capacitado para enviar (*palabras código*). Este proceso se llama *codificación*. Una vez codificado el mensaje lo enviamos a través del canal, y nuestro intermediario (el receptor) recibe un mensaje codificado (*palabra recibida*) posiblemente erróneo, ya que en todo proceso de comunicación hay ruido e interferencias. El mensaje recibido c' es traducido nuevamente a términos originales x', es decir, es *decodificado*. Todo el proceso se resume en el siguiente esquema



En general, $x' \neq x$ y es deseable que este error sea detectado (lo cual permite pedir una retransmisión del mensaje) y en lo posible corregido.

La *Teoría de Códigos Autocorrectores* se ocupa del segundo y cuarto pasos del esquema anterior, es decir, de la codificación y decodificación de mensajes, junto con el problema de detectar y corregir errores. Del problema más general, que es considerar todo el proceso, el diseño de canales a usar, etcétera, se ocupa la *Teoría de la Información*, inaugurada por el trabajo fundacional de Claude Shannon en 1948. No hay que confundir a la Teoría de Códigos Autocorrectores con la Criptografía, que también es parte de la Teoría de la Información. Sin

embargo, en la segunda, lo que importa es enviar un mensaje, a un receptor amigo, que sea secreto e inviolable para los demás (el enemigo); mientras que en la primera, lo que interesa es enviar un mensaje con la mayor eficiencia y verosimilitud posibles.

En la vida cotidiana, convivimos con muchos códigos aunque no nos demos cuenta. Por ejemplo, los más comunes son el código de barras, el ISBN usado en los libros y el código ASCII usado en las computadoras. Los primeros ejemplos de códigos usados en la práctica son el código Morse, usado en telegrafía desde el siglo XIX, y el sistema Braille para no-videntes. Además, cualquier artefacto tecnológico que transmita o almacene mensajes digitales, sonidos, imágenes, etcétera, involucra al menos un código. Ejemplos típicos de ello son las computadoras, los teléfonos celulares, las transmisiones por satélites, los CD's y DVD's, la televisión, etcétera.

Supongamos ahora la siguiente situación concreta. Tenemos un vehículo de exploración en la superficie de Marte, que llamaremos $Mar\theta$, que manejamos a control remoto desde la Tierra por medio de un canal que transmite impulsos eléctricos de dos voltajes distintos, que denotamos simplemente por 0 y 1, respectivamente. El vehículo se mueve de a un metro por vez, en una de las cuatro direcciones posibles: norte (N), sur (S), este (E) y oeste (O). Luego, nuestros mensajes son N, S, E y O y los codificamos, por ejemplo, como 00, 11, 01 y 10. Es decir,

$$\mathrm{N} \to 00, \qquad \mathrm{O} \to 10, \qquad \mathrm{S} \to 11, \qquad \mathrm{E} \to 01.$$

Ahora, supongamos que nuestro vehículo se encuentra orientado hacia el norte, al borde de un enorme cráter, con el precipicio a su derecha (o sea, hacia el este). Gráficamente,

$$\begin{array}{ccc} & \checkmark & \\ \checkmark & Mar\theta & \times \\ & \checkmark & \end{array}$$

Enviamos el mensaje 00, es decir "avance un metro hacia el norte". Una interferencia en la transmisión hace que $Mar\theta$ reciba 01, avance un metro a su derecha y se pierda para siempre en las profundidades del Planeta Rojo. Un error aquí es fatal, y nos cuesta millones de dólares... y el puesto.

El problema está en nuestro código

$$C_1 = \{00, 01, 10, 11\},$$

que *no detecta* errores. Es decir, si hay un error en la transmisión, la palabra recibida es otra palabra código. Más precisamente, si cometemos un error al enviar 00 recibimos 01 ó 10, y como ambas son palabras de C_1, no detectamos ningún error. Análogamente para 01, 10 y 11. Representemos este hecho por

$$00 \to \left\{ \begin{array}{l} 10 \in C_1 \\ 01 \in C_1 \end{array} \right. , \qquad 01 \to \left\{ \begin{array}{l} 00 \in C_1 \\ 11 \in C_1 \end{array} \right. , \qquad 10 \to \left\{ \begin{array}{l} 00 \in C_1 \\ 11 \in C_1 \end{array} \right. , \qquad 11 \to \left\{ \begin{array}{l} 01 \in C_1 \\ 10 \in C_1 \end{array} \right. .$$

Esto se debe a que C_1 consta de *todas* las palabras de longitud 2 que se pueden formar con ceros y unos.

¿Como podemos arreglar esto? La forma más fácil es agregar *redundancia* mediante un *dígito de control de paridad*, o sea, agregamos un dígito extra a cada palabra código de modo que la suma de los dígitos de cada palabra código sea par. En nuestro caso, el nuevo código es

$$C_2 = \{000, 011, 101, 110\} \subset \mathbb{Z}_2^3.$$

Si ahora transmitimos 000 y un error es cometido en la transmisión, entonces recibimos 100, 010 ó 001. Como ninguna de estas palabras pertenece al código, detectamos un error. Nuestro vehículo no se mueve y por lo tanto retransmitimos el mensaje esperando tener mejor suerte.

Lo mismo sucede con las otras palabras del código, es decir, si se comete un único error al enviar *cualquier* palabra código, la palabra recibida no pertenece al código. En símbolos,

$$000 \to \begin{cases} 100 \notin C_2 \\ 010 \notin C_2 \\ 001 \notin C_2 \end{cases} \qquad 011 \to \begin{cases} 111 \notin C_2 \\ 001 \notin C_2 \\ 010 \notin C_2 \end{cases} \qquad 101 \to \begin{cases} 001 \notin C_2 \\ 111 \notin C_2 \\ 100 \notin C_2 \end{cases} \qquad 110 \to \begin{cases} 010 \notin C_2 \\ 100 \notin C_2 \\ 111 \notin C_2 \end{cases}$$

Decimos entonces que el código C_2 *detecta un error* o que es *1-corrector*. Observar que C_2 no detecta 2 errores. Es decir, si se cometen 2 errores, la palabra recibida estará en el código cualquiera sea la palabra código enviada (controlar).

Sin embargo, el código C_2 no corrige errores. Esto es, una vez detectado el error, no se puede decidir cuál fue la palabra código enviada. En nuestro ejemplo, supongamos que enviamos 000 y recibimos 010. Si bien $Mar\theta$ detecta el error, si quisiera tomar una decisión por sí mismo, éste no podría. En efecto, suponiendo un error, la palabra 010 puede ser decodificada como 110, como 000 ó como 011, todas palabras códigos. Luego, $Mar\theta$ solicita la retransmisión del mensaje, esperando recibir 110, 000 ó 011, con mayor probabilidad que 101.

¿Cómo podemos mejorar nuestra situación? ¿Es posible hacer que $Mar\theta$ decida por si mismo? Afortunadamente la respuesta es sí. Una solución fácil es agregar mayor redundancia, a costa de tener que transmitir más y perder un poco más tiempo. Formamos el código

$$C_3 = \{000000, 000111, 111000, 111111\}$$

repitiendo tres veces cada dígito del código original C_1. Ahora, si mandamos nuestro mensaje 000000 y se comete un error, cualquier palabra que nos llegue puede ser corregida. Por ejemplo,

$$00 \xrightarrow[\text{codificamos}]{} 000000 \xrightarrow[+\ 1\ \text{error}]{} \left\{ \begin{array}{c} 100000 \\ 010000 \\ 001000 \\ 000100 \\ 000010 \\ 000001 \end{array} \right\} \xrightarrow[\text{corregimos}]{} 000000 \xrightarrow[\text{decodificamos}]{} 00$$

Aquí, si enviamos 000000 y recibimos 000100 no sólo detectamos el error sino que podemos corregirlo. Intuitivamente, 000100 está "*más cerca*" de 000000 que de 000111, 111000 ó 111111. Luego, corregimos 000100 como 000000 y no como 000111 ya que es *más probable* cometer un error que cometer tres. Que se haya mandado 111000 ó 111111 es mucho más improbable. Este nuevo código C_3 detecta hasta dos errores y corrige uno (¿porqué?) Luego, hemos mejorado las propiedades detectoras y correctoras del código original C_1 y de C_2.

A veces no es posible pedir retransmisión de mensajes y es por eso que los códigos autocorrectores son tan útiles y necesarios. Ejemplos de esto se dan en la transmisión de fotografías desde el espacio tomadas por sondas espaciales, al escuchar discos compactos o al ver DVD's, al realizar ciertas transmisiones vía satélite, etcétera.

Uno de los objetivos centrales de la teoría de códigos autocorrectores es construir "buenos" códigos. Esto es, códigos que permitan codificar *muchos* mensajes (tamaño grande), que se puedan transmitir *rápida* y *eficientemente* (alta tasa de información), que *detecten* y *corrijan* simultáneamente la mayor cantidad de errores posibles (distancia mínima grande) y que haya algoritmos de decodificación *fáciles* y *efectivos*. Como es de suponer, estas metas son casi siempre contradictorias entre sí, y se trata entonces de encontrar un balance entre todos los parámetros involucrados.

Hay muchos tipos de códigos autocorrectores. La clasificación más básica, teniendo en cuenta la estructura del código, es la siguiente

$$\text{Códigos} \begin{cases} \text{de bloque} \begin{cases} \text{lineales} \begin{cases} \text{cíclicos} \\ \text{de convolución} \\ \text{otros} \end{cases} \\ \text{no-lineales} \begin{cases} \text{sobre anillos} \\ \text{otros} \end{cases} \end{cases} \\ \text{de longitud variable} \begin{cases} \text{de Huffman} \\ \text{otros} \end{cases} \end{cases}$$

Las familias más conocidas de estos códigos son

* Códigos lineales: de Hamming, de Hamming extendidos, simplex, de Golay, de Reed-Muller, de Goppa geométricos.
* Códigos cíclicos: BCH, de Reed-Solomon, de residuos cuadráticos, de Goppa clásicos.
* Códigos no-lineales: Hadamard, Kerdock, Justesen, Preparata.

Sin duda, los códigos cíclicos y los de convolución son los más importantes por su sencillez y utilidad. Estos códigos, no sólo poseen buenas propiedades generales y algoritmos eficientes de codificación y decodificación, sino que pueden ser implementados en computadoras a través de ciertos circuitos lineales llamados *shift-registers*. No es de extrañar entonces que estos sean uno de los más utilizados en la práctica. En este curso nos interesaremos exclusivamente por los códigos lineales y cíclicos en general, y sólo veremos en algún detalle los códigos de Hamming, de Golay y de Reed-Muller.

Por último, quiero agradecer a los organizadores de elENA III por haberme dado la posibilidad de dar este curso. La Teoría de Códigos es una teoría muy bonita, que involucra diversas ramas de la matemática tales como el álgebra, la geometría algebraica, geometrías finitas y combinatoria, teoría de números y curvas elípticas, entre otras. Es un area donde se puede hacer investigación teórica, pero que posee muchas aplicaciones concretas a la tecnología, y es por esto que creo que merece ser más divulgada y cultivada. ¡A codificar que se acaba el mundo!

1. Generalidades sobre Códigos

1.1. Definiciones básicas. Un *alfabeto* es un conjunto finito $\mathcal{A} = \{a_1, \ldots, a_q\}$. A los elementos de $\mathcal{A}$ se los llama *símbolos* y el número q es la *raíz* de $\mathcal{A}$. Una *n-cadena* o *palabra de longitud n* sobre $\mathcal{A}$ es una sucesión de n elementos de $\mathcal{A}$. En general, escribiremos a las palabras por yuxtaposición de símbolos, es decir

$$a = a_{i_1} a_{i_2} \ldots a_{i_n}, \qquad a_{i_k} \in \mathcal{A},$$

y decimos que a tiene *longitud n*. A veces, sin embargo, será conveniente usar la notación vectorial, y escribir $a = (a_{i_1}, a_{i_2}, \ldots, a_{i_n})$. Denotamos por $\mathcal{A}^n$ el conjunto de todas las n-cadenas y por $\mathcal{A}^*$ el conjunto de todas las palabras sobre $\mathcal{A}$, es decir $\mathcal{A}^* = \bigcup_{n\in\mathbb{N}} \mathcal{A}^n$.

Definición 1.1. Si $\mathcal{A} = \{a_1, \ldots, a_q\}$ es un alfabeto, un *código q-ario* sobre $\mathcal{A}$ es un subconjunto C de $\mathcal{A}^*$. Los elementos de C se llaman *palabras códigos* (codewords). El número $M = |C|$ se llama el *tamaño* del código. Si todas las palabras códigos tienen longitud fija n decimos que C es un *código de bloque* con parámetros (n, M) o que C es un *(n, M)-código*. Si C no es de bloque decimos que C es *de longitud variable*.

Un ejemplo de código de longitud variable es $C = \{0, 10, 101, 1110, 11111\}$. El ejemplo más famoso de este tipo es sin duda el código Morse. Como no nos ocuparemos de estos códigos, de ahora en adelante por *código* entenderemos *código de bloque*.

Sea C un código q-ario. Se dice que C es un código *binario, ternario* o *cuaternario* según sea $q = 2$, $q = 3$ ó $q = 4$, respectivamente. Los códigos binarios son los más comunes y los más viejos. Últimamente, ha habido gran interés en ciertos códigos cuaternarios ya que poseen algunas mejoras respecto de los binarios.

La *tasa de información* de un (n, M)-código q-ario se define por

$$R = R_q(C) = \frac{\log_q(M)}{n}.$$

Esta tasa da una idea del porcentaje de dígitos que guardan la información del mensaje original sobre el total de dígitos transmitidos. Se buscan códigos con tasa de información alta, digamos $R > \frac{2}{3}$ ó $R > \frac{3}{4}$.

Ejemplo 1.2. Sean $\mathcal{A}_2 = \{0, 1\}$ y $\mathcal{A}_3 = \{0, 1, 2\}$. Los códigos binarios (sobre $\mathcal{A}_2$)

$$C_1 = \{0, 1\}, \qquad C_2 = \{00, 01, 10\},$$
$$C_3 = \{000, 111\}, \qquad C_4 = \{000, 011, 101, 110\}.$$

tienen parámetros $(1, 2)$, $(2, 3)$, $(3, 2)$ y $(3, 4)$, respectivamente. Las correspondientes tasas de información están dadas por

$$R(C_1) = \log_2(2) = 1, \qquad R(C_2) = \log_2(3)/2 \approx 0{,}7925,$$
$$R(C_3) = \log_2(2)/3 = \tfrac{1}{3}, \qquad R(C_4) = \log_2(4)/3 = \tfrac{2}{3}.$$

Los códigos ternarios

$$C_5 = \{001, 010, 012, 021, 100, 101, 120, 221, 222\},$$
$$C_6 = \{00000, 11111, 22222\},$$

tienen parámetros $(3, 9)$ y $(5, 3)$, y tasas de información $R(C_5) = \log_3(9)/3 = \frac{2}{3}$ y $R(C_6) = \log_3(3)/5 = \frac{1}{5}$, respectivamente. □

Notar que si $\mathcal{A}_q \subset \mathcal{A}_r$, con $q < r$, todo código q-ario C sobre $\mathcal{A}_q$ puede pensarse como un código r-ario sobre $\mathcal{A}_r$, en cuyo caso los parámetros (n, M) de C no cambian pero su tasa de información empeora, ya que

$$R_r(C) = \frac{\log_r(M)}{n} < \frac{\log_q(M)}{n} = R_q(C).$$

Luego, si el alfabeto no está explicitado, consideramos al código C sobre $\mathcal{A}_q$ con el menor q posible, pues así tiene la máxima tasa de información.

1.2. Distancia de Hamming y peso. Sean x e y dos palabras de igual longitud sobre el mismo alfabeto $\mathcal{A}$. La *distancia de Hamming* entre x e y, denotada por $d(x, y)$, se define como el número de coordenadas en que x e y difieren, es decir $d : \mathcal{A}^n \times \mathcal{A}^n \to [0, n] \subset \mathbb{N}$, donde

$$d(x, y) = \#\{1 \le i \le n : x_i \neq y_i\}.$$

Por ejemplo, si $x = 10221$ e $y = 12211$, entonces $d(x, y) = 2$. Veamos que $(\mathcal{A}^n, d)$ es un espacio métrico.

Proposición 1.3. *La función d es una distancia en $\mathcal{A}^n$, es decir, satisface las siguientes propiedades*

(**D1**) $d(x, y) \ge 0$ y $d(x, y) = 0$ *si y sólo si* $x = y$ *(positiva definida),*

(**D2**) $d(x, y) = d(y, x)$ *(simetría),*

(**D3**) $d(x, z) \le d(x, y) + d(y, z)$ *(desigualdad triangular).*

Demostración. Las propiedades (**D1**) y (**D2**) son obvias. Veamos (**D3**). Sean $x = x_1 \dots x_n$, $y = y_1 \dots y_n$ y $z = z_1 \dots z_n$. Sea $T = \{i : x_i \neq z_i\}$. Luego $d(x, z) = |T|$. Como T es la unión disjunta de los conjuntos $U = \{i : x_i \neq z_i \text{ y } x_i = y_i\}$ y $V = \{i : x_i \neq z_i \text{ y } x_i \neq y_i\}$ se tiene que $d(x, z) = |U| + |V|$. Ahora, por la definición de $d(x, y)$ es inmediato que $|V| \leq d(x, y)$. Por otra parte, si $i \in U$ entonces $y_i = x_i \neq z_i$ y por lo tanto $|U| \leq d(y, z)$. Luego d es una distancia. □

Definición 1.4. Dado un código C se define la *distancia de* C, y se la denota por $d(C)$ ó d_C, como la menor distancia no-nula entre sus palabras código, es decir

$$d = d_C = \min_{\substack{x,y \in C \\ x \neq y}} d(x, y). \tag{1.1}$$

Un (n, M, d)-*código* es un código de longitud n, tamaño M y distancia d.

Dado $x \in \mathbb{F}_q^n$ ($\mathbb{F}_q$ = el cuerpo finito de q elementos) se define el *peso de* x, denotado por $w(x)$, como el número de coordenadas no-nulas de x, es decir $w(x) = \#\{1 \leq i \leq n : x_i \neq 0\}$. O sea, el peso de x es la distancia de x al $\mathbf{0} = 00 \dots 0$, esto es $w(x) = d(x, 0)$. Por ejemplo, $w(0120211) = 5$. Si C es un código, el *peso de* C se define por

$$w(C) = \min_{\substack{x \in C \\ x \neq 0}} w(x). \tag{1.2}$$

De las definiciones (1.1) y (1.2) anteriores, es claro que para todo $x, y \in \mathbb{F}_q^n$ se cumple

$$d(x, y) = w(x - y). \tag{1.3}$$

Ahora, si $x = x_1 x_2 \dots x_n$ e $y = y_1 y_2 \dots y_n$ son palabras binarias, se define la *intersección* de x e y como la palabra

$$x \cap y = (x_1 y_1, x_2 y_2, \dots, x_n y_n),$$

o sea $(x \cap y)_i = 1$ si y sólo si $x_i = 1$ e $y_i = 1$. Para $x, y \in \mathbb{F}_2^n$ se tiene la siguiente relación

$$d(x, y) = w(x) + w(y) - 2w(x \cap y) \tag{1.4}$$

la cual se verifica fácilmente (ejercicio).

Observación 1.5. La distancia de Hamming es la única métrica que usaremos. Sin embargo, no es la única posible y no siempre es la más adecuada. Un ejemplo de esto es cuando se trata de números telefónicos. En $\mathbb{Z}_{10}^3$ tenemos $d(263, 264) = d(263, 363) = 1$. En la práctica, si estos son los números internos de telefónos de FaMAF, en el primer caso le erramos a mi oficina por la del lado, mientras que en el segundo le erramos por un piso. Sería entonces más práctico usar una métrica que sea sensible a las posiciones de los dígitos.

También está el problema de la ubicación de los dígitos en el aparato, que es la siguiente

1	2	3
4	5	6
7	8	9
	0	

Por ejemplo, el 1 tiene como vecinos al 2, al 4 y al 5. Luego, si quiero marcar el número 61 son más probables los errores 62 ó 64 que los errores 68 ó 71. También habría que tener en cuenta que 0, 1 y 3 tienen 3 vecinos, que 7 y 9 tienen 4 vecinos, que 2, 4 y 6 tienen 5 vecinos y que el 5 tiene 8 vecinos.

A partir de un $[10, 8]$-código sobre $\mathbb{F}_{11}$ se puede construir un código de longitud 10 sobre $\mathbb{Z}_{10}$, no-lineal (¿porqué?), con más de 82 millones de palabras código (que sobran para codificar los números telefónicos de la Argentina; por ejemplo, el número de FaMAF es 351-4334051). Este código automáticamente detecta y corrige 1 error, por lo que habría muchas menos llamadas equivocadas. ¿Puede el lector imaginar porqué no se utiliza?... ¡Correcto!

1.3. Códigos lineales. Para codificar y decodificar de manera más práctica y eficiente es útil dotar al alfabeto $\mathcal{A}$ de cierta estructura algebraica. Es común considerar a $\mathcal{A}$ como un *cuerpo finito* aunque también se lo puede considerar como un *anillo*. De ahora en adelante, fijamos $\mathcal{A} = \mathbb{F}_q$, el cuerpo finito de q elementos (salvo explícita mención de lo contrario). Recordamos que $\mathbb{F}_q$ es único salvo isomorfismo y que $q = p^r$ para algún primo p y $r \in \mathbb{N}$. Si $q = p$, tenemos $\mathcal{A} = \mathbb{Z}_p$, el cuerpo de enteros módulo p. El conjunto de n-cadenas $\mathcal{A}^n$ es un espacio vectorial sobre $\mathbb{F}_q$ de dimensión n, que identificamos naturalmente con

$$\mathbb{F}_q^n = \{(x_1, \ldots, x_n) : \ x_i \in \mathbb{F}_q, 1 \leq i \leq n\}$$

mediante la asignación $x_1 x_2 \ldots x_n \longleftrightarrow (x_1, x_2, \ldots, x_n)$.

Definición 1.6. Un *código lineal q-ario de longitud n y rango k* es un subespacio $L \subset \mathbb{F}_q^n$ de dimensión k. En este caso decimos que L es un $[n,k]_q$-*código*. Si L tiene distancia d entonces decimos que L es un $[n,k,d]_q$-*código*. Cuando L es un código binario, es usual quitar a $q = 2$ de la notación.

Notar que el tamaño de un $[n,k]_q$-código C es $M = q^k$, pues $C \simeq \mathbb{F}_q^k$. En este caso la tasa de información es

$$R = \frac{\log_q(q^k)}{n} = \frac{k}{n}.$$

Dado $V = \mathbb{F}_q^n$ hay dos códigos lineales triviales, $\{\mathbf{0}\}$ y V, con parámetros $[n, 0, -]$ y $[n, n, 1]$ respectivamente (notar que no se define la distancia para el código $\{\mathbf{0}\}$.) Veamos otros ejemplos.

Ejemplo 1.7. El *código de repetición q-ario*

$$Rep_q(n) = \{\underbrace{0 \ldots 0}_{n}, \underbrace{1 \ldots 1}_{n}, \ldots, \underbrace{(q-1) \ldots (q-1)}_{n}\}$$

es un $[n, 1, n]$-código lineal (ejercicio). □

Ejemplo 1.8. Los códigos C_1, C_3, C_4 y C_6 del Ejemplo 1.2 son lineales, mientras que C_2 y C_5 no lo son, ya que por ejemplo $01 + 10 = 11 \notin C_2$ y $010 + 012 = 022 \notin C_5$. Además, $C_1 = \mathbb{Z}_2$, $C_3 = Rep_2(3)$ y $C_6 = Rep_3(5)$. □

Ejemplo 1.9. El conjunto de todas las palabras de peso par en $\mathbb{F}_2^n$,

$$E(n) = \{x \in \mathbb{F}_2^n : w(x) \equiv 0 \,(\mathrm{mod}\ 2)\}, \tag{1.5}$$

es un código lineal binario con parámetros $[n, n-1, 2]$ (ejercicio). Por ejemplo, tenemos

$$E(3) = \{000, 011, 101, 110\}, \qquad E(4) = \{0000, 0011, 0101, 0110, 1001, 1010, 1100, 1111\},$$

con parámetros $[3, 2, 2]$ y $[4, 3, 2]$, respectivamente. □

Para calcular la distancia mínima de un (n, M)-código hacen falta calcular $\binom{M}{2} = \frac{M(M-1)}{2}$ distancias de Hamming. Sin embargo, la siguiente proposición nos dice que si el código es lineal podemos hacerlo sólo calculando $M - 1$ pesos.

Proposición 1.10. *Si C es un código lineal entonces $d(C) = w(C)$.*

Demostración. Como C es lineal, tenemos

$$d(C) = \min_{x \neq y \in C} d(x, y) = \min_{x \neq y \in C} w(x - y) = \min_{0 \neq x \in C} w(x) = w(C),$$

ya que $x - y$ recorre todos las palabras código de C cuando x e y recorren todo C. □

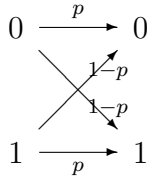
Ejemplo 1.11. El código $C = \{000, 011, 101, 110\}$ es lineal y por lo tanto $d_C = w_C = 2$. Si el código no es lineal, la proposición anterior no vale. Por ejemplo, el código $C' = \{11, 12, 21, 22\}$ tiene $d_{C'} = 1 < 2 = w_{C'}$ y el código $C'' = \{01, 10\}$ tiene $d_{C''} = 2 > 1 = w_{C''}$. □

Observación 1.12. Sea C un (n, M, d)-código. Los números n, M y d, son los parámetros básicos de C. Por otra parte, la tasa de información R y el número $\delta = d/n$, son parámetros secundarios y tienen que ver con la eficiencia de C durante la transmisión de mensajes. En general, fijado un n, interesan códigos con M grande (para transmitir muchos mensajes distintos) y d grande (para que detecte y corrija el mayor número de errores). Veremos que estas metas son intrínsecamente contradictorias entre sí, y se busca entonces un buen balance entre los parámetros. Existen muchas cotas que los parámetros de un código deben cumplir y un problema central en la teoría es construir códigos óptimos en el sentido que realicen alguna de dichas cotas.

1.4. Canales y decodificación. Sean $\mathcal{A} = \{a_1, \ldots, a_q\}$ y $\mathcal{B} = \{b_1, \ldots, b_r\}$ dos alfabetos, con $\mathcal{A} \subset \mathcal{B}$. Un *canal discreto aleatorio* es un canal de comunicación que envía símbolos de $\mathcal{A}$ y recibe símbolos en $\mathcal{B}$, con probabilidad $p(b_j|a_i)$ de recibir el símbolo b_j dado que se envió el símbolo a_i. Luego, para cada $1 \leq i \leq q$, se tiene

$$\sum_{j=1}^{r} p(b_j|a_i) = 1.$$

Es común suponer que se trabaja con un *canal simétrico*, es decir $\mathcal{A} = \mathcal{B}$ y todos los símbolos tienen iguales probabilidades de acierto $p(a_i|a_i) = p$, y de error $p(a_i|a_j) = \frac{1-p}{q-1}$, donde $i \neq j$ y $1 \leq i \leq q$. Por ejemplo, para un canal simétrico binario, esquemáticamente tenemos



Siempre asumiremos que transmitimos con un *canal discreto aleatorio simétrico*. También supondremos que los errores aparecen al azar durante la transmisión, es decir que todas las coordenadas de las palabras recibidas tienen la misma probabilidad de error.

Ahora, sea $C \subset \mathcal{A}^n$ un código q-ario y supongamos que al transmitir la palabra código $c \in C$ recibimos la palabra $x \notin C$, ¿cómo decodificamos x? Existen muchas estrategias posibles. Lo más sensato es asignarle a x la palabra código c que sea *más probable*, es decir,

$$p(x|c) = \max_{y \in C} p(x|y),$$

donde

$$p(x|y) = \prod_{i=1}^{n} p(x_i|y_i)$$

es la probabilidad de recibir x dado que se envió y. Este método se llama *decodificación por máxima verosimilitud* (maximum likelihood decoding).

Otra forma igualmente válida es asignarle a x la palabra código c que sea *más cercana*. Es decir, si

$$d(x, c) = \min_{y \in C} d(x, y),$$

donde d es una distancia en C (por ejemplo la de Hamming), entonces decodificamos a x por c. A ésta se la conoce como *decodificación por distancia mínima* (minimum distance decoding).

Bajo la hipótesis de un canal simétrico y tomando d como la distancia de Hamming, se puede ver que la decodificación por distancia mínima y por máxima verosimilitud son *equivalentes*.

1.5. Equivalencia de códigos. Existen, en la literatura, varias nociones de equivalencia entre códigos. Adoptaremos la siguiente. Dos códigos q-arios (sobre el mismo alfabeto) son equivalentes si uno puede ser obtenido del otro por permutaciones de coordenadas y símbolos, es decir por una combinación de operaciones del siguiente tipo:

(**C**) permutaciones de las *coordenadas* en el código,

(**S**) permutaciones de los *símbolos* en una posición fija (o en varias).

Ejemplo 1.13. $C = \{00100, 00011, 11111, 11000\}$ y $C' = \{00000, 01101, 10110, 11011\}$ son códigos equivalentes, pues C' se obtiene de C intercambiando las coordenadas 2 con 4 (operación de tipo (**C**)) y luego intercambiando los símbolos 0 y 1 en la tercer coordenada (operación de tipo (**S**)). En efecto,

$$C = \left\{ \begin{array}{c} 0\underline{0}1\underline{0}0 \\ 0\underline{0}0\underline{1}1 \\ 1\underline{1}1\underline{1}1 \\ 1\underline{1}0\underline{0}0 \end{array} \right\} \longrightarrow \left\{ \begin{array}{c} 00\overline{1}00 \\ 01\overline{0}01 \\ 11\overline{1}11 \\ 10\overline{0}10 \end{array} \right\} \longrightarrow \left\{ \begin{array}{c} 00000 \\ 01101 \\ 11011 \\ 10110 \end{array} \right\} = C'.$$

□

La definición formal es la siguiente

Definición 1.14. Dos (n, M)-códigos q-arios C_1 y C_2 son *equivalentes*, y se denota por $C_1 \simeq C_2$, si existe una permutación $\sigma \in \mathbb{S}_n$ de las n coordenadas y permutaciones $\pi_1, \ldots, \pi_n \in \mathrm{Biy}(\mathcal{A})$ del alfabeto, tales que

$$c_1 c_2 \ldots c_n \in C_1 \quad \Leftrightarrow \quad \pi_1(c_{\sigma(1)})\pi_2(c_{\sigma(2)}) \ldots \pi_n(c_{\sigma(n)}) \in C_2.$$

Observación 1.15. Es claro que, si $C_1 \simeq C_2$, entonces $(n_1, M_1, d_1) = (n_2, M_2, d_2)$ y, por lo tanto, C_1 y C_2 corrigen el mismo número de errores. Luego, suponiendo un canal simétrico, el rendimiento de códigos equivalentes es idéntico en términos de corrección de errores.

Observación 1.16. Todo (n, M, d)-código C, sobre un alfabeto $\mathcal{A}_q$ que contiene al 0, resulta equivalente a un código C' que contiene la palabra nula $\mathbf{0} = 00\ldots0$. En efecto, si $\mathbf{0} \notin C$, elijo cualquier $x \in C$. Si $i_1, \ldots, i_k$ son las coordenadas no-nulas de x, tomamos la permutación $\pi = (0x_{i_n}) \ldots (0x_{i_2})(0x_{i_1})$ (identificando $\mathrm{Biy}(\mathcal{A}_q) \simeq \mathbb{S}_q$). Es claro que $C' = \pi(C)$ contiene a $\mathbf{0}$.

Una definición alternativa es la siguiente. Dos (n, M)-códigos C y C' sobre $\mathcal{A} = \mathbb{F}_q$ son *múltiplo escalar equivalentes* si C' se obtiene de C aplicando operaciones de tipo

(**C**) permutaciones de las coordenadas en el código,

(**M**) *multiplicación* de los símbolos en una coordenada fija, o en varias, por un *escalar* no-nulo $\alpha \in \mathbb{F}_q^*$.

Notar que en el caso binario no hay operaciones de tipo (**M**) no triviales. Como $\mathcal{A} = \mathbb{F}_q$ es un cuerpo, si dos códigos son múltiplo escalar equivalentes entonces son equivalentes ya que si $\alpha \in \mathbb{F}_q^*$, la aplicación $x \mapsto \alpha x$ es una biyección de $\mathbb{F}_q$. La recíproca no es cierta como lo muestra el siguiente ejemplo.

Ejemplo 1.17. El código $C = \{012, 120, 201\}$ es equivalente al código ternario de repetición $Rep_3(3) = \{000, 111, 222\}$. En efecto, aplicando las permutaciones $\pi_2 = (021)$ y $\pi_3 = (012)$ en la segunda y tercera coordenada respectivamente tenemos

$$C = \left\{ \begin{array}{c} 012 \\ 120 \\ 201 \end{array} \right\} \xrightarrow{\pi_2} \left\{ \begin{array}{c} 002 \\ 110 \\ 221 \end{array} \right\} \xrightarrow{\pi_3} \left\{ \begin{array}{c} 000 \\ 111 \\ 222 \end{array} \right\} = Rep_3(3).$$

Sin embargo C y $Rep_3(3)$ no son múltiplo escalar equivalentes (¿porqué?) □

1.6. Esferas y códigos perfectos. Dado $x \in \mathcal{A}^n$, con $|\mathcal{A}| = q$ y $r \geq 0$, se define la *esfera de radio r centrada en x* como

$$S_q(x,r) = \{y \in \mathcal{A}^n : d(x,y) = r\}$$

y la *bola de radio r centrada en x* como

$$B_q(x,r) = \{y \in \mathcal{A}^n : d(x,y) \leq r\} = \bigcup_{i=0}^{r} S_q(x,i).$$

Se define el volumen $V_q(n,r)$ como el cardinal de cualquier bola de radio r en $\mathcal{A}^n$. Luego,

$$V_q(n,r) = |B_q(x,r)| = \sum_{i=0}^{r} |S_q(x,r)| = \sum_{i=0}^{r} \tbinom{n}{i}(q-1)^i. \tag{1.6}$$

Veamos que las bolas de radio $t = \lfloor \frac{d_C-1}{2} \rfloor$ centradas en palabras códigos son disjuntas.

Lema 1.18. *Si C es un código con distancia mínima $d_C = 2t+1$ ó $d_C = 2t+2$, entonces*

$$B_q(c,t) \cap B_q(c',t) = \emptyset$$

para todo $c, c' \in C$ con $c \neq c'$.

Demostración. Sea $x \in B_q(c,t)$ con $c \in C$. Entonces, $x \notin B_q(c',t)$ para todo $c' \in C$ con $c' \neq c$. Si no fuera así, por la desigualdad triangular tendríamos

$$d(c,c') \leq d(c,x) + d(x,c') \leq t + t = 2t < 2t+1 = d_C,$$

lo cual es absurdo. □

Esto permite probar la siguiente cota de Hamming, también llamada *cota de empaquetamiento de esferas*.

Proposición 1.19 (Cota de Hamming)**.** *Si C es un $(n, M, d)_q$-código con $d = 2t+1$ ó $d = 2t+2$ entonces*

$$M \cdot \sum_{i=0}^{t} \tbinom{n}{i}(q-1)^i \leq q^n. \tag{1.7}$$

Demostración. Como las bolas de radio $t = \lfloor \frac{d-1}{2} \rfloor$ son disjuntas, y cada bola $B_q(c,t)$ con $c \in C$ contiene $V_q(n,t)$ palabras de $\mathcal{A}^n$, resulta la desigualdad (1.7). □

Para códigos *lineales* q-arios con parámetros $[n,k,d]$ la cota de Hamming toma la expresión

$$\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \tbinom{n}{i}(q-1)^i \leq q^{n-k}, \tag{1.8}$$

y, en particular, para códigos lineales *binarios* se reduce a

$$\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \tbinom{n}{i} \leq 2^{n-k}. \tag{1.9}$$

Ejemplo 1.20. La cota de Hamming da una cota superior para el tamaño M que un código de longitud n y distancia d puede tener. Por ejemplo, si C es un $[6,k,3]$-código binario entonces, como C es lineal y $t = 1$, tenemos

$$M = 2^k \leq \frac{2^6}{1 + \binom{6}{1}} = \frac{64}{7} < 10.$$

Luego, $k \leq 3$. Es decir, no existen códigos lineales con parámetros $[6,k,3]$ y $4 \leq k \leq 6$. □

Definición 1.21. Un código $C \subset \mathcal{A}^n$ se dice *perfecto* si existe un r tal que las bolas de radio r centradas en las palabras códigos son todas disjuntas entre sí y cubren todo el espacio, es decir

$$\mathcal{A}^n = \dot{\bigcup_{c \in C}} B_q(c, r).$$

Ejemplo 1.22. Consideremos un código famoso, el *código de Hamming binario de longitud* 7.

$$\mathcal{H}_2(3) = \left\{ \begin{array}{cccc} 0000000, & 0001101, & 1111111, & 1110010, \\ 1101000, & 1000110, & 0010111, & 0111001, \\ 0110100, & 0100011, & 1001011, & 1011100, \\ 0011010, & 1010001, & 1100101, & 0101110 \end{array} \right\}. \tag{1.10}$$

Es fácil verificar que $\mathcal{H}_2(3)$ es un código lineal con parámetros $[7, 4, 3]$. Como $d = 3 = 2 \cdot 1 + 1$, las bolas de radio $r = 1$, centradas en palabras códigos, son todas disjuntas. Ahora,

$$|\mathcal{H}_2(3)| = 16, \qquad |B_2(c, 1)| = 1 + \tbinom{7}{1} = 8, \qquad |\mathcal{A}^7| = 2^7 = 128.$$

Como $16 \cdot 8 = 128$, deducimos que las bolas $B_2(c, 1)$, con $c \in \mathcal{H}_2(3)$, cubren todo $\mathcal{A}^7$. Luego, $\mathcal{H}_2(3)$ es un código perfecto. □

El tamaño de un código perfecto está determinado por su longitud y su distancia mínima. El siguiente resultado se conoce con el nombre de *condición de empaquetamiento de esferas*.

Teorema 1.23. *Sea C un $(n, M, d)_q$-código. Entonces C es perfecto si y sólo si $d = 2t + 1$ y*

$$M \cdot \sum_{k=0}^{t} \tbinom{n}{k} (q-1)^k = q^n. \tag{1.11}$$

Nota. El teorema dice que un código es perfecto si y sólo si la distancia es impar y se alcanza la igualdad en la cota de Hamming (1.7).

Demostración. Supongamos que C es perfecto. Si $d = 2t + 2$ es par, las esferas de radio $r \leq t$ son disjuntas pero no cubren $\mathcal{A}^n$. Por otra parte, las esferas de radio $r = t + 1$ cubren $\mathcal{A}^n$ pero no son disjuntas (ejercicio). Luego $d = 2t + 1$ es impar. Las esferas de radio t son disjuntas y cubren $\mathcal{A}^n$, luego se cumple la igualdad en (1.7) y, por lo tanto, vale (1.11).

Recíprocamente, si vale (1.11) para un $(n, M, 2t + 1)$-código, como las esferas de radio t son disjuntas y cubren todo $\mathcal{A}^n$, entonces C es perfecto. □

Observación 1.24. Es importante notar que la existencia de números n, M y t que satisfagan (1.11) no implica la existencia de un código perfecto con parámetros $(n, M, 2t+1)$. Por ejemplo, los números $n = 90, M = 2^{78}$ y $t = 2$ satisfacen la condición del empaquetamiento de esferas para $q = 2$. En efecto,

$$\sum_{k=0}^{2} \tbinom{90}{k} = 1 + 90 + \tbinom{90}{2} = 1 + 90 + 45 \cdot 89 = 4096 = 2^{12},$$

y como $2^{78} \cdot 2^{12} = 2^{90}$, entonces (1.11) vale.

Existen algunas relaciones entre la teoría de códigos y la teoría de diseños en combinatoria. Por ejemplo, hay un resultado que dice que si C es un (n, M, d)-código binario perfecto entonces los números

$$\lambda_s = \frac{\binom{n-s}{t+1-s}}{\binom{2t+1-s}{t+1-s}}$$

son *enteros* para todo $1 \leq s \leq t$. Volviendo a nuestro ejemplo, tenemos que

$$\lambda_1 = \frac{\binom{89}{2}}{\binom{4}{2}} = \frac{4005}{6} = \frac{1335}{2} \notin \mathbb{Z}, \qquad \text{y} \qquad \lambda_2 = \frac{\binom{88}{1}}{\binom{3}{1}} = \frac{88}{3} \notin \mathbb{Z}.$$

Luego, no existe un código binario perfecto con parámetros $(90, 2^{78}, 5)$.

Es fácil chequear (ejercicio) que las siguientes familias de parámetros (n, M, d) satisfacen la condición de empaquetamiento de esferas (1.11):

(1) $(n, q^n, 1)$,
(2) $(n, 1, 2n+1)$,
(3) $(2m+1, 2, 2m+1)$,
(4) $(\frac{q^r-1}{q-1}, q^{n-r}, 3)$, $\qquad r \geq 2$,
(5) $(23, 2^{11}, 7)$
(6) $(11, 3^6, 5)$.

Los parámetros en (1) corresponden al código $\mathcal{A}^n$ con $|\mathcal{A}| = q$, y los de (3) al código de repetición binario de longitud impar $Rep_2(2m+1)$. Para el código trivial de longitud n, $\{\mathbf{0}\}_n$, la distancia no está definida. Si convenimos es tomar $d = 2n+1$, entonces los parámetros en (2) corresponden a $\{\mathbf{0}\}_n$. Estos son los llamados *códigos perfectos triviales*.

Los códigos perfectos resultan muy interesantes por la gran simetría con que se encuentran distribuídas las palabras códigos. Surgen naturalmente las siguientes preguntas:

Pregunta 1.12. ¿Existen códigos con los parámetros dados en (4), (5) y (6) más arriba?

Sí. En la Sección 3, veremos que los códigos de Hamming $\mathcal{H}_q(r)$ y los códigos de Golay pinchados $\mathcal{G}_{23}$ y $\mathcal{G}_{11}$, tienen los parámetros dados en (4), (5) y (6), respectivamente. Luego, hay códigos perfectos no-triviales. Sin embargo, estos son todos lineales.

Pregunta 1.13. ¿Hay códigos perfectos no-lineales?

Sí. Durante algún tiempo se conjeturó que los únicos códigos perfectos no triviales eran los de Hamming y los de Golay. Sin embargo, en la Observación 3.5 mostraremos como construir un código no-lineal con los parámetros de Hamming y, por lo tanto, perfecto.

Pregunta 1.14. ¿Están clasificados los códigos perfectos?

No por el momento, aunque existen resultados parciales para códigos sobre alfabetos cuyo cardinal es potencia de un primo. El siguiente resultado fue probado por Tietäväinen (1973), siguiendo grandes contribuciones de van Lint, e independientemente también por Zinov'ev y Leont'ev (1973).

Teorema 1.25. *Sea C un código perfecto no-trivial q-ario, donde q es potencia de un primo. Entonces, C tiene los mismos parámetros que un código de Hamming o de Golay, es decir*

$$(\tfrac{q^r-1}{q-1}, q^{n-r}, 3), \qquad (23, 2^{11}, 7) \qquad \acute{o} \qquad (11, 3^6, 5).$$

Más aún,

(1) *Si C tiene los parámetros de Golay, es equivalente al correspondiente código de Golay.*

(2) *Si C es* lineal *y tiene los parámetros de Hamming, entonces es equivalente al código de Hamming correspondiente.*

Queda sin resolver el problema general, es decir la clasificación de los códigos perfectos sobre alfabetos de cardinal q arbitrario. Otro problema, por (2) del Teorema 1.25, es encontrar todos los códigos perfectos no-lineales.

1.7. Detección y corrección de errores. Si al transmitir una palabra código se cometen t errores, es decir, si la palabra enviada y la recibida difieren en exactamente t coordenadas, decimos que *se cometió un error de peso* t. Supongamos que $C \subset \mathcal{A}^n$ es un código q-ario sobre $\mathcal{A}$, con $\mathcal{A}$ un grupo abeliano. Si transmitimos $c \in C$ y recibimos $x \in \mathcal{A}^n$ con $d(x,c) = t$, entonces existe $e \in \mathcal{A}^n$ tal que

$$x = c + e \qquad \text{y} \qquad w(e) = t,$$

es decir $e = x - c$. Se dice que e es un *patrón de error*.

Definición 1.26. Se dice que un código *detecta s errores* si cuando en una palabra código se comete un error de peso r, con $1 \leq r \leq s$, la palabra resultante no es una palabra código. Un código es *s-detector* si detecta s errores pero no detecta $s+1$ errores (es decir, hay al menos un error de peso $s+1$ que el código no detecta).

Se dice que un código *corrige t errores* si, al decodificar por distancia mínima, se pueden corregir todos los errores de peso t o menos. Un código es *t-corrector* si corrige t errores pero no corrige $(t+1)$-errores.

Ejemplo 1.27. En la Introducción, vimos que el código $C_1 = \{00, 01, 10, 11\}$ no detecta ningún error, que $C_2 = \{000, 011, 101, 110\}$ detecta 1 error, aunque no lo puede corregir y, finalmente, que $C_3 = \{000000, 000111, 111000, 111111\}$ es un código 2-detector y también 1-corrector. □

Intuitivamente, cuánto más separadas se encuentren las palabras código entre sí, más fácil será detectar o corregir errores. En efecto, La capacidades detectoras y correctoras de un código están completamente determinadas por su distancia mínima.

Teorema 1.28. *Sea $C \subset \mathcal{A}^n$ un código con distancia mínima d.*

(i) *C es s-detector si y sólo si $d = s + 1$.*

(ii) *C es t-corrector si y sólo si $d = 2t + 1$ ó $d = 2t + 2$.*

Demostración. (i) Supongamos que C es s-detector y que $d \leq s$. Sean $c, c' \in C$ tales que $d(c, c') \leq s$. El error formado por las coordenadas en que c y c' difieren tiene peso menor que s y no es detectado por C, absurdo. Luego, $d \geq s + 1$. Si $d = s + t$ con $t \geq 1$, la recíproca en (i) implica que C es $(s + t - 1)$-corrector y por lo tanto $t = 1$, o sea $d = s + 1$.

Recíprocamente, sea $c \in C$ y $x \in \mathcal{A}^n$. Si $d(c, x) = s < d$, entonces $x \notin C$ y, por lo tanto, C detecta errores de peso $s = d - 1$ o menos. Ahora, si $c, c' \in C$ son tales que $d(c, c') = d$, el error formado por las coordenadas en que c y c' difieren tiene peso $d = s + 1$ y no es detectado por C. Luego, C es s-detector.

(ii) Supongamos que $d = 2t + 1$ ó $d = 2t + 2$. Por el Lema 1.18, sabemos que las bolas de radio t centradas en palabras códigos son disjuntas. Luego, al decodificar por distancia mínima, C corrige errores de peso t o menos. Si $d = 2t + 1$, existen $c, c' \in C$ con $d(c, c') = 2t + 1$, es decir, c y c' difieren en $2t + 1$ coordenadas. Supongamos que enviamos c y recibimos x con exactamente $t + 1$ errores, localizados en las coordenadas antedichas, y que x coincide con c' en esas $t + 1$ coordenadas. Como $d(x, c) = t + 1$, y $d(x, c') = 2t + 1 - (t + 1) = t$, al decodificar por distancia mínima, decodificamos *incorrectamente* a x como c'. Luego, C no es $(t+1)$-corrector. Si $d = 2t + 2$, la demostración es similar y se deja como ejercicio.

Ahora, supongamos que C es t-corrector, entonces $d \geq 2t + 1$ (¿porqué?) Por otra parte, si $d \geq 2t + 3 = 2(t + 1) + 1$, entonces, por el argumento previo, C es $(t + 1)$-corrector, absurdo. Luego, $d = 2t + 1$ ó $d = 2t + 2$. □

Corolario 1.29. *Si un código C tiene distancia mínima d, entonces C es $(d-1)$-detector y $\lfloor \frac{d-1}{2} \rfloor$-corrector.*

Sabemos que si C es un (n, M, d)-código, entonces o bien C detecta $d-1$ errores, o bien C corrige $\lfloor \frac{d-1}{2} \rfloor$ errores. ¿Qué sucede si queremos usar un mismo código para detectar y corregir errores de manera simultánea? Si queremos maximizar las propiedades de corrección, se pierde un poco en la detección de errores. Tenemos el siguiente resultado sobre estrategias mixtas que dejamos como ejercicio para el lector aplicado.

Teorema 1.30. *Un código C es simultáneamente t-corrector y $(t+s)$-detector si y sólo si $d = 2t+s+1$.*

Esto sirve cuando la distancia mínima de C es par, o sea $d = 2t+2$. En este caso, el código C simultáneamente corrige t errores y detecta $t+1$ errores. Sin embargo, si la distancia es impar no ganamos nada.

1.8. Construcciones. Damos a continuación algunos métodos comúnmente utilizados para obtener nuevos códigos a partir de otros ya dados.

Extensión (extending a code). El proceso de agregar una o más coordenadas a las palabras de un código se conoce como *extensión del código*. La forma más común de extender un código es agregando un dígito de chequeo de paridad total (*overall parity check digit*). Si C es un (n, M, d)-código, el *código extendido* $\hat{C}$ se define como

$$\hat{C} = \{c_1 c_2 \dots c_n c_{n+1} : \ c_1 c_2 \dots c_n \in C \quad \text{y} \quad \sum_{k=1}^{n+1} c_k = 0\}.$$

Es decir, tomamos $c_{n+1} = -c_1 - \dots - c_n$. Así, $\hat{C}$ es un $(\hat{n}, \hat{M}, \hat{d})$-código con

$$\hat{n} = n+1, \qquad \hat{M} = M, \qquad \hat{d} = d \ \text{ ó } \ d+1.$$

Luego, si bien el código extendido no mejora las cualidades para *corregir* errores al menos tiene una mejor capacidad para *detectar* errores.

Notar que si C es lineal entonces $\hat{C}$ también lo es (ejercicio) y que si C es binario entonces $\hat{C} \subset E(n+1)$ (ver (1.5)), es decir $\hat{C}$ tiene todas sus palabras de peso par.

Ejemplo 1.31. Extendiendo el código binario $C = \{00, 01, 10, 11\} = \mathbb{Z}_2^2$ se obtiene el código $\hat{C} = \{000, 011, 101, 110\}$. Notar que $d = 1$ pero $\hat{d} = 2$. Sin embargo, extendiendo una vez más tenemos $\hat{\hat{C}} = \{0000, 0110, 1010, 1100\}$, luego $\hat{\hat{d}} = \hat{d} = 2$. En general, si $C = \mathbb{Z}_2^n$ entonces $\hat{C} = E(n+1)$ (ejercicio). □

Pinchado (puncturing a code). El proceso opuesto a extender un código se denomina *pinchado de un código* en el que una o más coordenadas son quitadas de las palabras códigos. Si C es un (n, M, d)-código q-ario, con $d \geq 2$, entonces el *código pinchado* C^*, obtenido pinchando una de las coordenadas de C, tiene parámetros

$$n^* = n-1, \qquad M^* = M, \qquad d^* = d \ \text{ ó } \ d-1.$$

Notar que si C es lineal, C^* también resulta lineal (ejercicio).

Ejemplo 1.32. El código $\mathcal{H}_2(3)$ visto en el Ejemplo 1.22 tiene parámetros $[7, 4, 3]$. El código pinchado $\mathcal{H}_2^*(3)$ debe tener parámetros $[6, 4, 3]$ ó $[6, 4, 2]$. En el Ejemplo 1.20 vimos que no existen códigos con parámetros $[6, 4, 3]$, por lo que $\mathcal{H}_2^*(3)$ es un $[6, 4, 2]$-código. □

Ejemplo 1.33. El $(23, 4096, 7)$-código de Golay $\mathcal{G}_{23}$ se obtiene pinchando el $(24, 4096, 8)$-código de Golay $\mathcal{G}_{24}$ en la última coordenada. Luego, pinchando un código que no es perfecto puede obtenerse un código perfecto. □

Para códigos binarios, el proceso de extender y pinchar códigos sirve para probar lo siguiente.

Proposición 1.34. *Existe un $(n, M, 2t+1)$-código binario si y sólo si existe un $(n+1, M, 2t+2)$-código binario.*

Demostración. Sea C un $(n, M, 2t+1)$-código binario. Como cada palabra código en $\hat{C}$ tiene peso par, (1.4) implica que la distancia entre dos palabras códigos de $\hat{C}$ es par, luego $d_{\hat{C}} = 2t+2$.

Recíprocamente, supongamos que C es un $(n+1, M, 2t+2)$-código binario y sean $c, c' \in C$ tal que $d(c, c') = 2t+2$. Si pinchamos el código C en una coordenada en que c y c' difieren, el código C^* resultante tiene distancia mínima $2t+1$. □

Como una aplicación sencilla, pinchando un código $d-1$ veces consecutivas se obtiene la siguiente cota

Proposición 1.35 (Cota de Singleton)**.** *Si C es un (n, M, d)-código q-ario entonces*

$$M \leq q^{n-d+1}.$$

En particular, si C es lineal vale $k \leq n-d+1$.

Demostración. Sea $f : \mathbb{F}_q^n \to \mathbb{F}_q^{n-d+1}$ la aplicación que borra las últimas $d-1$ coordenadas (o sea, $f(C) = C^{*(d-1)}$). Es claro que $f(C)$ es un código en $\mathbb{F}_q^{n-d+1}$. Como $f_{|C}$ es 1-1, pues sino habría $c, c' \in C$ tales que $d(c, c') \leq d-1$, entonces $|f(C)| = |C|$, y por lo tanto $M \leq q^{n-d+1}$. □

Expurgado (expunging or expurgate). *Expurgar* un código es eliminar algunas palabras del código. Como ejemplo, sea L un (n, M, d)-código lineal binario. Si L contiene al menos una palabra de peso impar entonces la mitad de las palabras de L son de peso impar (ejercicio). Tirando las palabras de peso impar obtenemos un $(n, M/2, d')$-código con $d' \geq d$. Si d es impar entonces $d' > d$.

Aumentado (augmenting). El proceso opuesto a expurgar un código es *aumentar un código*, es decir agregar palabras adicionales al código. Un modo común de aumentar un código binario C es incluir los complementos de cada palabra código en C. El *complemento* x^c de una palabra binaria x es la palabra que se obtiene intercambiando los 0's por los 1's. Sea C^c el conjunto de los complementos de las palabras en C. Claramente, si $x, y \in \mathbb{F}_2^n$ entonces $d(x, y^c) = n - d(x, y)$.

Proposición 1.36. *Sea C un (n, M, d)-código binario. Entonces*

$$d_{C \cup C^c} = \min\{d, n - d_{\max}\}$$

donde $d_{\max}$ es la distancia máxima entre las palabras códigos de C.

Demostración. Tenemos

$$d_{C \cup C^c} = \min\left\{d_C, d_{C^c}, \min_{c \in C, c' \in C^c} d(c, d)\right\}.$$

Notar que $d_C = d_{C^c} = d$ y que por el lema anterior

$$\min_{c \in C, d \in C^c} d(c, d) = \min_{c, d \in C} d(c, d^c) = \min_{c, d \in C}\{n - d(c, d)\} = n - \max_{c, d \in C} d(c, d)$$

y el resultado sigue. □

Sea L un código lineal binario. Si $\mathbf{1} = 11\ldots1 = \mathbf{0}^c \in L$ entonces $L = L^c$. Sin embargo , si $\mathbf{1} \notin L$, entonces $L \cap L^c = \emptyset$ (ejercicio). Por lo visto, el siguiente resultado es claro.

Proposición 1.37. *Si L es un (n, M, d)-código lineal binario que no contiene a la palabra $\mathbf{1}$, entonces $L \cup L^c$ es un $(n, 2M, d')$-código lineal binario con $d' = \min\{d, n - w_{\max}\}$ donde $w_{\max}$ es el peso máximo de las palabras de L.*

Acortamiento (shortening). *Acortar* un código significa quedarse sólo con las palabras códigos que tienen un cierto símbolo en una cierta coordenada (por ejemplo, 0 en la primer coordenada) y luego borrar esa coordenada. Si C es un (n, M, d)-código entonces el *código acortado* tiene longitud $n-1$ y distancia mínima d. El código acortado que se obtiene tomando s en la coordenada i se llama la *sección* $x_i = s$ (cross-section) de C, y lo denotamos por $C_{\{x_i=s\}}$. La prueba del siguiente resultado se deja como ejercicio.

Proposición 1.38. *Si C es un (n, M, d)-código lineal binario entonces la sección $x_i = 0$ es un $(n-1, \frac{1}{2}M, d)$-código lineal binario.*

Ejemplo 1.39. Dejamos como ejercicio ver que $\mathbb{F}_q^n{}_{\{x_1=0\}} = \mathbb{F}_q^{n-1}$ y $E(n)_{\{x_1=0\}} = E(n-1)$. □

Suma directa. Si C_1 es un $(n_1, M_1, d_1)_q$-código y C_2 es un $(n_2, M_2, d_2)_q$-código, la *suma directa* de C_1 y C_2 es el código

$$C_1 C_2 = \{c_1 c_2 : c_1 \in C_1, c_2 \in C_2\},$$

es decir, la yuxtaposición de los códigos. Es claro que C_1C_2 tiene parámetros

$$n = n_1 + n_2, \qquad M = M_1 M_2, \qquad d = \min\{d_1, d_2\}.$$

Si $C_2 = C_1 = C$ escribimos C^2 en lugar de CC. En general tenemos la *potencia* C^m de C. Si C es un $(n, M, d)_q$-código entonces C^m es un $(mn, mM, d)_q$-código.

Producto tensorial. Si C_1 es un $[n_1, k_1, d_1]_q$-código y C_2 es un $[n_2, k_2, d_2]_q$-código, el *producto tensorial (o de Kronecker)* de C_1 y C_2 es el $[n_1n_2, k_1k_2, d_1d_2]$-código

$$C_1 \otimes C_2 = \{c_1 \otimes c_2 : c_1 \in C_1, c_2 \in C_2\}.$$

O sea, $C_1 \otimes C_2$ puede pensarse como el código cuyas palabras códigos consisten en todas las matrices $n_1 \times n_2$ cuyas filas pertenecen a C_1 y cuyas columnas pertenecen a C_2 (el conjunto de estas matrices puede ser identificado con un subespacio de $\mathbb{F}_q^{n_1 n_2}$).

Por supuesto se puede considerar el producto tensorial de un número finito de códigos. En particular, el producto tensorial $C^{\otimes m}$ de C es un $[n^m, k^m, d^m]_q$-código.

Ejemplo 1.40. Si $C = E(3) = \{000, 011, 101, 110\}$ entonces $C^{\otimes 2} = \{c \otimes c' : c, c' \in C\}$ es el código formado por las matrices:

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

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

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

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

□

La construcción $(u, u+v)$. Esta es una construcción muy útil. Sólo puede realizarse con códigos de la misma longitud y sobre el mismo alfabeto $\mathbb{F}_q$. Sea $C_1 \subset \mathbb{F}_q^n$ un (n, M_1, d_1)-código y $C_2 \subset \mathbb{F}_q^n$ un (n, M_2, d_2)-código. Se define el código

$$C_1 \oplus C_2 = \{c(c+d) : c \in C_1, d \in C_2\}.$$

Proposición 1.41. *El código $C_1 \oplus C_2$ tiene parámetros $(2n, M_1M_2, d')$ con $d' = \min\{2d_1, d_2\}$.*

Demostración. Es claro que $C_1 \oplus C_2$ tiene longitud $2n$ y tamaño $M_1 M_2$. Veamos la distancia. Sean $u_1 = c_1(c_1 + d_1)$ y $u_2 = c_2(c_2 + d_2)$ dos palabras códigos distintas . Si $d_1 = d_2$ entonces

$$d(u_1, u_2) = 2d(c_1, c_2) \geq 2d_1.$$

Por otra parte, si $d_1 \neq d_2$ entonces

$$d(u_1, u_2) = w(u_1 - u_2) = w(c_1 - c_2) + w(c_1 - c_2 + d_1 - d_2) \geq w(d_1 - d_2) = d(d_1, d_2) \geq d_2.$$

Luego, $d_{C_1 \oplus C_2} \geq \text{mín}\{2d_1, d_2\}$. Como la igualdad se puede dar en todos los casos, la proposición sigue. □

Pegado (pasting). Sean C_1 y C_2 códigos lineales con parámetros $[n_1, k, d_1]_q$ y $[n_2, k, d_2]_q$, respectivamente, definidos por los mapas $\phi_1 : \mathbb{F}_q^k \to \mathbb{F}_q^{n_1}$ y $\phi_2 : \mathbb{F}_q^k \to \mathbb{F}_q^{n_2}$ (ver Sección 2.1). Consideremos el mapa diagonal

$$(\phi_1, \phi_2) : \mathbb{F}_q^k \to \mathbb{F}_q^{n_1} \oplus \mathbb{F}_q^{n_2} = \mathbb{F}_q^{n_1+n_2}, \qquad x \mapsto (\phi_1 x, \phi_2 x).$$

A la imagen $C = \text{Im}(\phi_1, \phi_2)$ se la llama el *pegado* de C_1 y C_2. Claramente, el código C tiene parámetros $[n_1 + n_2, k, d_1 + d_2]_q$.

Aplicando esta construcción m-veces al mismo $[n, k, d]_q$-código C obtenemos la *repetición m-veces* del código C, denotado por mC, con parámetros $[mn, k, md]_q$.

Si C_1 y C_2 tienen matrices generadoras G_1 y G_2 (ver Sección 2.1) entonces el pegado tiene matriz generadora $G = (G_1|G_2)$ (ejercicio).

Intercalación (interleaving). Sea C_1 un $(n_1, k_1, d_1)_q$-código y C_2 un $(n_2, k_2, d_2)_q$-código, ambos sobre el mismo alfabeto. Supongamos que en cada código fijamos un orden en sus palabras, digamos

$$C_1 = \langle c_{11}, c_{12}, \ldots, c_{1M_1} \rangle, \qquad C_2 = \langle c_{21}, c_{22}, \ldots, c_{1M_2} \rangle,$$

donde $\langle , \rangle$ denotan un conjunto ordenado. Luego, podemos intercalar las palabras códigos de C_1 y C_2 para formar el *código intercalado*

$$C_1 \odot C_2 = \{c_{11}c_{21}, c_{12}c_{22}, \ldots, c_{1M}c_{2M}\}$$

donde $M = \text{mín}\{M_1, M_2\}$. Se puede ver que $C_1 \odot C_2$ tiene parámetros

$$(n_1 + n_2, \text{mín}\{M_1, M_2\}, d)$$

donde $d \geq d_1 + d_2$. Si algún C_i es vacío, entonces por definición $C_1 \odot C_2$ es el otro código.

Combinando las dos construcciones anteriores, pegado e intercalado, tenemos el siguiente resultado.

Proposición 1.42. *Sean C_1 y C_2 códigos sobre el mismo alfabeto con parámetros $(n_1, k_1, d_1)_q$ y $(n_2, k_2, d_2)_q$, respectivamente. Si $r, s \in \mathbb{N}$ entonces el código $rC_1 \odot sC_2$ tiene parámetros*

$$(rn_1 + sn_2, \text{mín}\{M_1, M_2\}, d), \qquad d \geq rd_1 + sd_1.$$

Observación 1.43. Existen otros tipos de intercalado, por ejemplo el *intercalado cruzado* (cross-interleaving), muy usados por la NASA y otras agencias espaciales, y que permiten que compacts discs rayados o muy usados funcionen igualmente bien (ver Apéndice).

Observación 1.44. Existen otros varios tipos de construcciones. Por ejemplo, la *restricción a un subcuerpo* $\mathbb{F}_{q'} \subset \mathbb{F}_q$ y la *concatenación*, en donde se cambia la raíz q del alfabeto. También existen la *extensión del alfabeto* y la *restricción del alfabeto*.

2. Códigos Lineales

Recordemos que un código lineal es un subespacio L de $\mathbb{F}_q^n$. Los códigos lineales son los códigos de bloque más simples y comunes. Hay una gran cantidad de familias de códigos lineales con buenas propiedades. Estos son fáciles de implementar, en particular los llamados cíclicos, y existen algoritmos generales de decodificación. Más adelante veremos, en algún detalle, los códigos de Hamming, de Golay y de Reed-Muller.

2.1. Matriz generadora. Comencemos estudiando una manera simple de generar el código.

Definición 2.1. Sea L un $[n,k]_q$-código. Una *matriz generadora de* L es una matriz $G \in \mathbb{F}_q^{k\times n}$ cuyas filas forman una base de L.

Notar que G siempre existe y tiene rango k. Observar que G *genera* L, es decir

$$L = \{uG : u \in \mathbb{F}_q^k\}. \tag{2.1}$$

En efecto, sea $c_1,\dots,c_k$ una base de L y $c_i = \sum_{j=1}^n c_{ij}e_j$ para ciertos $c_{ij} \in \mathbb{F}_q$, con $e_1,\dots,e_n$ la base canónica de $\mathbb{F}_q^n$. Si $u \in \mathbb{F}_q^k$, entonces $uG \in L$, pues

$$\begin{aligned} uG = (uG^1,\dots,uG^n) &= \Big(\sum_{i=1}^k u_ic_{i1},\dots,\sum_{i=1}^k u_ic_{in}\Big) \\ &= \sum_{j=1}^n\Big(\sum_{i=1}^k u_ic_{ij}\Big)e_j = \sum_{i=1}^k u_i\Big(\sum_{j=1}^n c_{ij}e_j\Big) = \sum_{i=1}^k u_ic_i. \end{aligned}$$

Recíprocamente, si $c \in L$, existen únicos $u_1,\dots,u_k$ tal que $c = \sum_{i=1}^k u_ic_i$. Luego, tomando $u = (u_1,\dots,u_k) \in \mathbb{F}_q^k$ se cumple que $c = uG$. En otras palabras, vimos que L es el *espacio fila* de G.

Además, $u_1G = u_2G$ si y sólo si $u_1 = u_2$. Es decir, distintos mensajes generan palabras códigos diferentes.

Si tenemos un código lineal L, buscamos una base de L y tenemos una matriz generadora G. Recíprocamente, si G es una matriz $k \times n$ en $\mathbb{F}_q$ de rango $k \le n$, entonces G genera el código $L = \mathbb{F}_q^kG \subset \mathbb{F}_q^n$.

La matriz generadora G da una forma fácil de codificar palabras de $\mathbb{F}_q^k$. Simplemente, a cada palabra $u \in \mathbb{F}_q^k$, la codificamos como $uG \in \mathbb{F}_q^n$.

Ejemplo 2.2. Sea $G = \left(\begin{smallmatrix}1&0&0\\0&1&1\end{smallmatrix}\right) \in M_{2\times3}(\mathbb{Z}_2)$. Como las filas son linealmente independientes, G tiene rango 2 y genera un código binario L con parámetros $[3,2]$. Como

$$(x_1,x_2)\begin{pmatrix}1&1&0\\0&1&1\end{pmatrix} = (x_1, x_1+x_2, x_2)$$

codificamos

$$00 \to 000, \qquad 01 \to 011, \qquad 10 \to 110, \qquad 11 \to 101.$$

Luego

$$L = \{000, 011, 101, 110\} = E(3)$$

y vemos que la distancia es $d = 2$. □

Consideremos la transformación lineal $R_G : \mathbb{F}_q^k \to \mathbb{F}_q^n$ dada por $u \mapsto uG$. Como R_G es 1–1, tenemos que $Im(R_G) = \mathbb{F}_q^kG = L$ y $L \simeq \mathbb{F}_q^k$. Hay k coordenadas que guardan la información y $n-k$ que son redundantes. Esta *redundancia* se utiliza para la detección y corrección de errores y en los algoritmos de decodificación. En el ejemplo anterior, vemos que cualquier par

de coordenadas guardan la información de los mensajes originales, y que la coordenada restante es redundante.

Observación 2.3. La matriz generadora también es útil para almacenar el código. Si L es un $[n,k]$-código, L consta de $M = q^k$ palabras códigos de longitud n. Luego, se necesitan nq^k dígitos q-arios (q-ary digits ¿*qits*?) para almacenar L. Sin embargo, todas estas palabras códigos se pueden obtener a partir de una matriz generadora G de L, es decir con kn dígitos q-arios. Por ejemplo, el código de Hamming $\mathcal{H}_2(4)$ es un $[15,11,3]$-código binario. Luego, hacen falta $15 \cdot 2^{11} = 30720$ *bits* (*b*inary dig*its*) para almacenar el código, contra $11 \cdot 15 = 165$ bits de la matriz generadora!

Definición 2.4. Un $[n,k]$-código q-ario es *sistemático* si existen k coordenadas $i_1, \ldots, i_k$ tal que al restringir las palabras código a estas coordenadas se obtienen todas las q^k palabras de longitud k.

Ejemplo 2.5. El código $C = \{000, 011, 101, 110\}$ es sistemático en las coordenadas 1 y 2. En realidad, C es sistemático en cualquier par de coordenadas (controlar). □

Si G genera L, entonces toda matriz reducida por filas de G genera el mismo código, ya que sólo cambia la base de L. Sin embargo, es mucho más fácil trabajar con la matriz escalón reducida por filas de G. Por ejemplo, la matriz

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

genera un $[4,3]$-código L dado por la transformación lineal

$$(x_1, x_2, x_3) \to (x_1 + x_3, x_1 + x_2, x_2, x_1 + x_2 + x_3).$$

Usando la matriz escalón reducida por filas de G,

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

el código está dado de manera más sencilla por

$$(x_1, x_2, x_3) \to (x_1, x_2, x_3, x_1 + x_3).$$

Luego, el código es

$$L = \{0000, 0011, 0100, 0111, 1001, 1010, 1100, 1111\},$$

y se ve claramente que es sistemático en las primeras tres coordenadas.

Todo $[n,k]$-código lineal L es sistemático en k coordenadas. En efecto, si L está generado por G, tomo G' la matriz escalón reducida por filas de G. Luego $L = \mathbb{F}_q^k G'$ es sistemático en los k coordenadas donde están los 1's líderes de G'.

Definición 2.6. Una matriz generadora se dice *en forma estándar* si es de la forma $G = (I_k|A)$ donde I_k es la matriz identidad $k \times k$ y A es $k \times n - k$.

Si G está en forma estándar, entonces L es sistemático en las k primeras coordenadas pues $uG = u(I_k|A) = (u|uA)$. En esta situación, codificar y decodificar es trivial ya que el esquema resulta

$$u \xrightarrow[\texttt{cod}]{} uG = (u|uA) \xrightarrow[\texttt{dec}]{} u \ .$$

Por otra parte, no todo código lineal tiene matriz generadora en forma estándar. Por ejemplo, si L está generado por $G = \left(\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$. Sin embargo, tenemos el siguiente resultado.

Proposición 2.7. *Todo código lineal L es equivalente a un código L' cuyo matriz generadora está forma estándar.*

Demostración. Si G es la matriz generadora de L, sea E la matriz escalón reducida por filas de G e $i_1, \dots, i_k$ las coordenadas donde están los 1's líderes de E. Si $E^1, \dots, E^n$ son las columnas de E, sea $\pi = (ki_k) \cdots (2i_2)(1i_1) \in \mathbb{S}_n$ y $G' = (E^{\pi(1)} E^{\pi(2)} \cdots E^{\pi(n)})$. Luego, se tiene $G' = (I_k|A)$ y $L' = \mathbb{F}_q^n G' \simeq L$. □

Corolario 2.8. *Dado un $[n,k]$-código L y $1 \leq i_1 \leq \cdots \leq i_k \leq n$, existe un código lineal L' equivalente a L, y sistemático en las coordenadas $i_1, \dots, i_k$.*

2.2. Código dual y matriz de control de paridad. El espacio vectorial $\mathbb{F}_q^n$ tiene un producto interno natural dado por

$$x \cdot y = x_1 y_1 + \cdots + x_n y_n \qquad x, y \in \mathbb{F}_q^n.$$

Definición 2.9. Si L es un $[n,k]_q$-código, el conjunto

$$L^\perp = \{x \in \mathbb{F}_q^n : x \cdot c = 0 \text{ para todo } c \in L\}$$

es el *código dual de L.*

Teorema 2.10. *Sea L un $[n,k]_q$-código.*

(i) Si G es una matriz generadora de L entonces

$$L^\perp = \{x \in \mathbb{F}_q^n : xG^\top = 0\} = \{x \in \mathbb{F}_q^n : Gx^\top = 0\}.$$

(ii) $L^\perp$ es un $[n, n-k]_q$-código.

(iii) $L^{\perp\perp} = L$.

Demostración. (i) Por definición, $x \in L^\perp$ si y sólo si $x \cdot c = 0$ para todo $c \in L$. Luego,

$$0 = x \cdot c = xc^\top = x(uG)^\top = (xG^\top)u^\top$$

para algún $u \in \mathbb{F}_q^k$. Si $xG^\top = 0$ entonces $x \in L^\perp$. Recíprocamente, si $x \in L^\perp$ entonces $(xG^\top)u^\top = 0$ para todo $u \in \mathbb{F}_q^k$. En particular, para $u = e_1, \dots, e_k$, los vectores de la base canónica. Luego, $0 = (xG^\top){e_i}^\top = (uG^\top)^i$ para $1 \leq i \leq k$. Por lo tanto $uG^\top = 0$.

(ii) Es claro que $L^\perp$ es un subespacio de $\mathbb{F}_q^n$. Por (i),

$$L^\perp = \{x \in \mathbb{F}_q^n : Gx^\top = 0\},$$

o sea $L^\perp$ es el espacio solución de k ecuaciones con n incógnitas. Luego, como G tiene rango k, hay $n-k$ variables libres, por lo tanto $\dim L^\perp = n-k$.

(iii) Notar que $L \subset L^{\perp\perp} = \{x \in \mathbb{F}_q^n : x \cdot c' = 0 \text{ para todo } c' \in L^\perp\}$. Pero

$$\dim L^{\perp\perp} = n - (n-k) = k = \dim L,$$

luego $L = L^{\perp\perp}$. □

Si W es un subespacio vectorial de un $\mathbb{R}$-espacio vectorial con producto interno V, entonces $W \cap W^\perp = \{0\}$. Para espacios vectoriales sobre cuerpos finitos esto no es cierto en general. Es decir, existen códigos lineales $L \subset \mathbb{F}_q^n$ tales que $L \cap L^\perp \neq \{0\}$. Por ejemplo, si tomamos el $[4,2]$-código $L = \{0000, 0011, 1100, 1111\}$ se tiene que $L \subset L^\perp$ y como $L^\perp$ también es un $[4,2]$-código, tenemos $L = L^\perp$.

Vimos que agregando un dígito de control de paridad a un $[n,k]_q$-código obtenemos un $[n+1,k]_q$-código. Es posible agregar varios dígitos de control de paridad, de modo que cada uno de éstos realice un control de paridad entre un cierto conjunto de coordenadas.

Por ejemplo, tomemos las palabras de $\mathbb{Z}_2^3$ y agreguemos 3 dígitos de control de paridad de la siguiente manera: el primero chequea las dos primeras coordenadas, el segundo chequea la primera y la tercera y el último chequea la segunda y la tercera. Es decir, si $x_1x_2x_3 \in \mathbb{Z}_2^3$, tomamos $x_1x_2x_3y_1y_2y_3 \in \mathbb{Z}_2^6$ donde

$$x_1 + x_2 = y_1, \quad x_1 + x_3 = y_2 \quad \text{y} \quad x_2 + x_3 = y_3.$$

Luego, el código es

$$L = \{000\,000, 001\,011, 010\,101, 011\,110, 100\,110, 101\,101, 110\,011, 111\,000\}.$$

Las ecuaciones anteriores son equivalentes al sistema

$$\text{(2.2)} \qquad \begin{cases} x_1 + x_2 + y_1 = 0 \\ x_1 + x_3 + y_2 = 0 \\ x_2 + x_3 + y_3 = 0 \end{cases} \qquad \text{cuya matriz es} \quad H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Equivalentemente, L es el conjunto solución de (2.2), o sea $L = \{x \in \mathbb{Z}_2^6 : Hx^\top = 0\}$.

En general, las ecuaciones como en (2.2) se llaman *ecuaciones de control de paridad* y H es la *matriz de paridad*. Veremos que todo código lineal tiene una matriz de paridad, es decir, puede ser obtenido como el espacio solución de un sistema lineal homogéneo formado por ciertas ecuaciones de control de paridad.

Definición 2.11. Sea L un $[n,k]_q$-código. Una matriz H se dice *matriz de paridad* de L si es una matriz generadora de $L^\perp$.

Observación 2.12. (1) H siempre existe y es una matriz $n-k \times n$.

(2) Se cumple $GH^\top = 0$. En efecto, sean $c \in L, c' \in L^\perp$, entonces $c = uG$ y $c' = wH$ para ciertos $u \in \mathbb{F}_q^k, w \in \mathbb{F}_q^{n-k}$. Luego, $c \cdot c' = 0$ si y sólo si

$$0 = uG \cdot wH = uG(wH)^\top = u(GH^\top)w^\top$$

lo que a su vez sucede si y sólo si $GH^\top = 0$, pues $e_i(GH^\top){e_j}^\top = (GH^\top)_{ij}$.

(3) Recíprocamente, si H es $n-k \times n$ y $GH^\top = 0$ entonces H genera $L^\perp$, por lo tanto es una matriz de paridad de L.

(4) Si G es una matriz generadora de L entonces G es una matriz de paridad de $L^\perp$. Esto es así pues la matriz de paridad de $L^\perp$ es $H^\perp = (G^\perp)^\perp = G$.

Proposición 2.13. *Sea H la matriz de paridad de un $[n,k]_q$-código L. Entonces,*

$$L = \{x \in \mathbb{F}_q^n : xH^\top = 0\} = \{x \in \mathbb{F}_q^n : Hx^\top = 0\}.$$

Demostración. Si $c \in L$, entonces $c = uG$ donde $u \in \mathbb{F}_q^k$ y G es la matriz generadora de L. Luego $cH^\top = uGH^\top = 0$ y por lo tanto $L \subset S_H = \{x \in \mathbb{F}_q^n : Hx^\top = 0\}$, el espacio solución de un sistema de $n-k$ ecuaciones con n incógnitas y rango $n-k$. Como $\dim S_H = n - (n-k) = k = \dim L$, tenemos que $L = \{x \in \mathbb{F}_q^n : Hx^\top = 0\}$. □

Por lo visto hasta aquí, podemos ver a un código lineal L a través de transformaciones lineales. Como una imagen, $L = \operatorname{Im} R_G$, donde G es una matriz generadora de L, o como un núcleo, $L = \operatorname{Ker} R_{H^\top}$, donde H es una matriz de paridad de L. Esto da una forma alternativa de definir códigos lineales. Tener un código lineal q-ario de longitud n y rango k es equivalente a tener una sucesión exacta de la forma

$$0 \longrightarrow \mathbb{F}_q^k \xrightarrow{R_G} \mathbb{F}_q^n \xrightarrow{R_{H^\top}} \mathbb{F}_q^{n-k} \longrightarrow 0,$$

donde G y H son matrices $k \times n$ y $n-k \times n$, respectivamente, de rango máximo.

¿Cómo obtenemos H a partir del código L? Una forma sencilla es a través de G. Si $G = (I_k|A)$ está en forma estándar, entonces $H = (-A^\top|I_{n-k})$ es una matriz de paridad de L. En efecto, tenemos

$$GH^\top = (I_k|A)\begin{pmatrix} -A \\ I_{n-k} \end{pmatrix} = -A + A = 0,$$

ya que $G_i(H^\top)^j = (e_i|A_i)\cdot(-A^j|e_j) = -e_iA^j + A_ie_j = -a_{ij} + a_{ij} = 0$. Tal H se dice *matriz de paridad en forma estándar*, aunque no está en forma estándar como matriz generadora de $L^\perp$.

Un código lineal L se dice *auto-ortogonal* si $L \subset L^\perp$ y *autodual* si $L = L^\perp$. Un código autodual tiene parámetros $[2m, m]$. Si L es un código autodual, toda matriz generadora de L es matriz de paridad y recíprocamente. Luego, si $G = (Id_m \mid A)$ es una matriz generadora de L, entonces $H = (-A^\intercal \mid Id_m)$ también lo es.

2.3. Distancia de un código lineal y algunas cotas. No hay una manera efectiva de calcular la distancia d_L de un código lineal L a partir de una matriz generadora. Sin embargo, esto es posible a partir de una matriz de paridad.

Teorema 2.14. *Sea L un $[n, k, d]_q$-código y H una matriz de paridad de L. Entonces*

$$d = \text{mín}\{r : \text{ hay } r \text{ columnas linealmente dependientes en } H\}.$$

O sea, H tiene d columnas linealmente dependientes, pero cualquier conjunto de $d-1$ columnas son linealmente independientes.

Demostración. Sean $H^1, \ldots, H^n$ las columnas de H.

$$c \in L \Leftrightarrow cH^\top = 0 \Leftrightarrow c_1(H^\top)_1 + \cdots c_n(H^\top)_n = 0 \Leftrightarrow c_1H^1 + \cdots + c_nH^n = 0.$$

Ahora, si $c \in L$, c tiene peso r si y sólo si hay un conjunto de r columnas linealmente dependiente en H. Como $r \geq d$, no puede haber $d-1$ columnas linealmente dependientes en H. □

Este teorema puede usarse para construir códigos lineales con distancia d prefijada.

Como consecuencia del teorema anterior se obtienen 2 cotas, muy importantes y conocidas, que deben cumplir los parámetros de un $[n, k, d]_q$-código. La primera es muy sencilla y se la conoce como cota de Singleton.

Proposición 2.15 (Singleton)**.** *Si L es un $[n, k, d]_q$-código entonces*

$$d \leq n - k + 1.$$

Demostración. La matriz de paridad H es $n-k \times n$ y por el teorema anterior cualquier conjunto de $d-1$ columnas de H son linealmente independientes, luego $d - 1 \leq n - k$. □

Ejemplo 2.16. Un $[11, 6, d]_q$-código tiene $d \leq 11 - 6 + 1 = 6$. Un $[11, k, 7]_q$-código tiene $k \leq n - d + 1 = 5$. Un $[n, 8, 7]_q$-código tiene $n \geq d + k - 1 = 14$. □

Ejemplo 2.17. No existen $[n, n-1, 3]_q$-códigos. □

Los códigos cuyos parámetros alcanzan la igualdad en la cota de Singleton se llaman *MDS* (por "maximum distance separable" en inglés), ya que tienen la mayor distancia posible, dados una longitud y un tamaño fijos. Para $q = 2$, los únicos códigos MDS son triviales, es decir $Rep_2(n)$, $E(n)$ y $\mathbb{Z}_2^n$ con parámetros $[n, 1, n]$, $[n, n-1, 2]$ y $[n, n, 1]$, respectivamente. Luego, interesan los códigos MDS q-arios para $q > 2$.

La importancia del próximo teorema es que es un resultado de existencia. Si los números n, k, d y q cumplen cierta desigualdad, entonces existe un código con esos parámetros. Más aún, la prueba da un método para construirlo.

Proposición 2.18 (Gilbert-Varshamov)**.** *Sean $n, k, d, q \in \mathbb{N}$, con q potencia de un primo. Si se cumple la desigualdad*

$$q^{n-k} > \sum_{i=0}^{d-2} \tbinom{n-1}{i}(q-1)^i \tag{2.3}$$

entonces existe un $[n,k]_q$-código L con distancia $d_L \geq d$.

Demostración. Veamos que si se cumple (2.3) entonces podemos definir una matriz H de tamaño $n-k \times k$ y rango $n-k$ (que será la matriz de paridad del código) tal que cualquier conjunto de $d-1$ columnas es linealmente independiente.

Como primer columna tomamos cualquier vector no-nulo $h_1 \in \mathbb{F}_q^{n-k}$. Como segunda columna tomamos cualquier vector $h_2 \in \mathbb{F}_q^{n-k} \setminus \{0\}$ que no sea múltiplo de h_1. En general, queremos elegir la i-ésima columna $h_i \neq 0$ de modo que $h_i \notin \langle h_{i_1}, \ldots, h_{i_{d-2}} \rangle$ con $1 \leq i_1 \leq \cdots \leq i_{d-2} \leq i-1$. O sea, que h_i no sea combinación lineal de cualquier conjunto de $d-2$ columnas previamente elegidas. Pero el número de combinaciones lineales de $d-2$ columnas o menos, de las $i-1$ existentes es:

$$N_i = \tbinom{i-1}{1}(q-1) + \tbinom{i-1}{2}(q-1)^2 + \cdots + \tbinom{i-1}{d-2}(q-1)^{d-2} = \sum_{j=1}^{d-2} \tbinom{i-1}{j}(q-1)^j.$$

Luego, hay $q^{n-k} - N_i - 1$ elecciones para la i-ésima columna h_i de H. Así, si $q^{n-k} - N_n - 1 > 0$ podemos completar H. Por Teorema 2.14, H es la matriz de paridad de un $[n,k,d']_q$ código con $d' \geq d$. □

En particular, si $q = 2$ el teorema dice que si se cumple la desigualdad

$$2^{n-k} > \sum_{i=0}^{d-2} \tbinom{n-1}{i}$$

entonces existe un $[n,k,d]$-código binario.

Ejemplo 2.19. Sea $n = 7$ y $d = 3$. Luego, $2^{7-k} > \binom{6}{0} + \binom{6}{1} = 7$ se cumple si $k \leq 4$. Por lo tanto, Gilbert-Varshamov asegura la existencia de códigos lineales binarios con parámetros $[7,k,3]$ con $1 \leq k \leq 4$.

¿Existe un $[7,5,3]$-código? Como la cota de Singleton se cumple, no sabemos. Ahora, como la cota de Gilbert-Varshamov no se cumple, tampoco sabemos nada. Además, como en este caso $\lfloor \frac{d-1}{2} \rfloor = 1 = d-2$, la cota de Hamming coincide con la de Gilbert-Varshamov y no nos da información extra. □

Ejemplo 2.20. Sea $k = 5$ y $d = 3$. Por la cota de Singleton sabemos que $n \geq 7$. La desigualdad $2^{n-5} > \binom{n-1}{0} + \binom{n-1}{1} = n$ se cumple para $n = 9$ y no antes. Por Gilbert-Varshamov, existe un $[9,5,3]$-código binario, dado por ejemplo por la matriz

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

□

2.4. Decodificación por síndrome. Veamos ahora un método general de decodificación para códigos lineales $L \subset \mathbb{F}_q^n$, debido a Slepian (1960), que saca provecho de la estructura de espacio vectorial del espacio cociente $\mathbb{F}_q^n/L$.

Recordemos que $\mathbb{F}_q^n/L = \{x + L : x \in \mathbb{F}_q^n\}$ es el espacio vectorial formado por las coclases $x + L = \{x + c : c \in L\}$, con las operaciones

$$a(x + L) = ax + L, \qquad (x + L) + (y + L) = (x + y) + L,$$

donde $a \in \mathbb{F}_q$, $x, y \in \mathbb{F}_q^n$. El número de coclases es $|\mathbb{F}_q^n/L| = |\mathbb{F}_q^n|/|L| = q^{n-k}$.

Sea L un $[n, k]_q$-código con matriz de paridad H. Si $x \in \mathbb{F}_q^n$, el *síndrome de* x se define por

$$s(x) = s_H(x) := xH^\intercal.$$

Notar que $x \in L$ si y sólo si $s(x) = 0$. Más aún, x e y tienen el mismo síndrome si y sólo si yacen en la misma coclase. En efecto,

$$x + L = y + L \Leftrightarrow x - y \in L \Leftrightarrow (x - y)H^\intercal = 0 \Leftrightarrow xH^\intercal = yH^\intercal.$$

Es decir, el síndrome es un invariante de las coclases.

Veamos ahora como utilizar esto para decodificar por distancia mínima (que equivale a decodificación por máxima verosimilitud usando canales simétricos aleatorios). Supongamos que una palabra código es enviada y recibimos la palabra $x \in \mathbb{F}_q^n$. Luego, x se decodifica como una palabra código c a distancia mínima de x. O sea,

$$d(x, c) = \min_{c' \in L} d(x, c').$$

Podemos escribir $x = c + a$, donde $c \in L$ y $x, a \notin L$. Luego, buscar c a distancia mínima de x es equivalente a buscar $a = x - c$ con peso mínimo, ya que

$$\min_{c \in L} d(x, c) = \min_{c \in L} w(x - c) = \min_{a \in x + L} w(a).$$

Por lo tanto, como cuando c recorre el código L, a recorre la coclase $x + L$, la decodificación por distancia mínima requiere encontrar la palabra de peso mínimo entre todas las palabras de una coclase. Resumiendo, tenemos el siguiente resultado.

Teorema 2.21. *Sea L un código lineal con matriz de paridad H. Decodificar por distancia mínima es equivalente a decodificar la palabra recibida x como la palabra código $c = x - a$ donde a es una palabra de peso mínimo en la coclase $x + L$, o equivalentemente, donde a es una palabra de peso mínimo con igual síndrome que x.*

Este proceso de decodificación se realiza con el llamado *arreglo estándar* o *arreglo Slepiano* para L

$$\begin{array}{ccccc} 0 & c_1 & c_2 & \cdots & c_r \\ a_1 & c_1 + a_1 & c_2 + a_1 & \cdots & c_r + a_1 \\ a_2 & c_1 + a_2 & c_2 + a_2 & \cdots & c_r + a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_s & c_1 + a_s & c_2 + a_s & \cdots & c_r + a_s \end{array}$$

donde $r = q^k - 1$ y $s = q^{n-k} - 1$. La primera fila consta del código, o sea de la coclase $0 + L$. Para formar la segunda fila, tomamos una palabra de peso mínimo a_1 que no esté en la primera fila. Esto da la coclase $a_1 + L$. En general, para formar la i-ésima fila, tomamos a_i de peso mínimo que no se encuentre en las primeras $i - 1$ filas. Este proceso termina cuando escribimos q^{n-k} filas. Las palabras a_i se llaman *líderes de coclases* (*coset leaders*) del arreglo. Luego, dos palabras tienen igual síndrome si y sólo si están en la misma fila del arreglo.

Supongamos que recibimos la palabra x que está en la columna j del arreglo, entonces $x = c_j + a_i$ para cierto a_i, donde a_i es una palabra de peso mínimo en la coclase $x + L$. Luego, decodificamos x como c_j, es decir, como la palabra código de la columna j.

Ejemplo 2.22. Sea L el $[4, 2]$-código binario dado por la matriz generadora

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

Las coclases son

$$\begin{aligned} 0000 + L &= \{0000, 0100, 1101, 1001\} \\ 1000 + L &= \{1000, 1100, 0101, 0001\} \\ 0010 + L &= \{0010, 0110, 1111, 1011\} \\ 1010 + L &= \{1010, 1110, 0111, 0011\} \end{aligned}$$

Como elegimos los líderes de las coclases de peso mínimo, el arreglo queda

0000	0100	1101	1001
1000	1100	0101	0001
0010	0110	1111	1011
1010	1110	0111	0011

Si recibimos la palabra $x = 0111$, detectamos que hay un error pues x no está en la primer fila del arreglo. Luego, ésta es corregida como la palabra código $c = 1101$, que está arriba en la misma columna que x. □

Por fortuna, no es necesario almacenar todo el arreglo. Sólo necesitamos guardar una tabla con los líderes de las coclases y sus correspondientes síndromes, ya que cada fila esta determinada por éstos. Si recibimos la palabra x, calculamos su síndrome $s(x)$ y buscamos en la tabla el líder de coclase a_i con igual síndrome que x. Luego, decodificamos x como $c = x - a_i$. Esta es la llamada *decodificación por síndrome*.

Ejemplo 2.23. Sea L el código del ejemplo anterior. Haciendo operaciones elementales a G obtenemos una matriz generadora en forma estándar $G' = \left(\begin{smallmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{smallmatrix}\right)$ y por lo tanto la matriz de paridad es

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

La tabla de líderes y síndromes queda

Líderes	Síndromes
0000	00
1000	01
0010	10
1010	11

Ahora, si por ejemplo recibimos $x = 1110$, calculamos su síndrome

$$s(x) = 1110 \cdot H^{\intercal} = 11.$$

Luego, según la tabla, decodificamos x como $c = 1110 + 1010 = 0100$. □

Es importante notar que los errores en la transmisión que este método corrige son los que tienen el patrón de error de los líderes de las coclases y sólo esos. En efecto, supongamos que c es la palabra código que fue transmitida y $x = c + e$ es la palabra recibida donde e es el error. Si e es un líder de coclases, entonces x está en la fila liderada por e, y así al decodificar x obtenemos $x - e = c$, la palabra código correcta. Por otra parte, si e no es un líder de coclases y está en la

fila j del arreglo, entonces x también está en la fila j y es decodificada *incorrectamente* como $x - a_j \neq x - e = c$.

Observación 2.24. Si L tiene distancia mínima d, entonces todas las palabras $x \in \mathbb{F}_q^n$ de peso a lo sumo $t = \lfloor \frac{d-1}{2} \rfloor$ son líderes de coclases. En efecto, supongamos que dos palabras $x \neq y$ de peso a lo sumo t están en la misma coclase. Luego, por desigualdad triangular tenemos $d(x, y) \leq w(x) + w(y) \leq 2t \leq d - 1$, lo cual es absurdo.

La observación anterior permite realizar la siguiente estrategia conocida como *decodificación incompleta*, especialmente apropiada cuando la distancia es par. Si $d_C = 2t + 1$ ó $d_C = 2t + 2$, este método asegura la corrección de todos los errores de peso a lo sumo t en todas las palabras códigos y además, en algunos casos, permite detectar errores de peso mayores a t.

Dividimos el arreglo estándar en dos partes. En la parte superior está el código y todas las coclases con líderes de peso menor o igual a t. En la parte inferior están el resto de las coclases, es decir aquellas con líderes con peso mayor a t. Si la palabra recibida x está en la parte superior, decodificamos de la forma usual. Si x está en la parte inferior, detectamos que se han cometido al menos $t + 1$ errores y pedimos retransmisión del mensaje.

Ejemplo 2.25. Sea C el código binario generado por $G = \left(\begin{smallmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{smallmatrix}\right)$. Luego, el arreglo queda

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

Si recibimos 11110, lo decodificamos como 10110; pero si recibimos 10011, entonces pedimos retransmisión. □

Como antes, no es necesario almacenar todo el arreglo y uno puede quedarse con la tabla formada por las coclases y sus síndromes. Más aún, basta hacer esta tabla para la parte superior del arreglo.

Ejemplo 2.26. La matriz de paridad del código C del ejemplo anterior es $H = \left(\begin{smallmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{smallmatrix}\right)$ y la tabla de líderes y síndromes queda

Líderes	Síndromes
00000	000
10000	110
01000	011
00100	100
00010	010
00001	001

Si recibimos 11001, calculamos el síndrome $11001 \cdot H^\intercal = 100$ y por lo tanto decodificamos $11001 + 00100 = 11101$. Pero si recibimos 01110, como el síndrome $01110 \cdot H^\intercal = 101$ no está en la tabla, pedimos retransmisión. □

2.5. Enumeradores de peso y la identidad de MacWilliams. Si C es un (n, M)-código, denotamos con A_k el número de palabras código de peso k, es decir

$$A_k = \#\{c \in C : w(c) = k\}. \tag{2.4}$$

Los números $A_0, \dots, A_n$ se conocen como la *distribución de pesos* de C y la suma formal

$$W_C(s) = \sum_{k=0}^{n} A_k s^k \tag{2.5}$$

es el *enumerador de peso de* C.

Enunciamos ahora, sin demostración, una identidad muy importante y conocida que relaciona el enumerador de peso de un código lineal L con el de su dual $L^\perp$.

Teorema 2.27 (Identidad de MacWilliams)**.** *Sea $L \subset \mathbb{F}_q^n$ un código lineal, $L^\perp$ su dual, y*

$$W_L(s) = \sum_{k=0}^{n} A_k s^k, \qquad W_{L^\perp}(s) = \sum_{k=0}^{n} A_k^\perp s^k,$$

los enumeradores de peso de L y $L^\perp$, respectivamente. Entonces,

$$W_{L^\perp}(s) = \frac{1}{|L|}\big(1 + (q-1)s\big)^n \, W_L\left(\tfrac{1-s}{1+(q-1)s}\right). \tag{2.6}$$

Ejemplo 2.28. Consideremos el código binario $L = \{000, 111\}$. Usemos la identidad de MacWilliams para calcular $L^\perp$. Como $A_0 = A_3 = 1$ y $A_1 = A_2 = 0$, tenemos

$$W_L(s) = 1 + s^3,$$

y por (2.6) vale

$$W_{L^\perp}(s) = \tfrac{1}{2}(1+s)^3\Big(1 + \big(\tfrac{1-s}{1+s}\big)^3\Big) = \tfrac{1}{2}\big((1+s)^3 + (1-s)^3\big) = 1 + 3s^2.$$

Luego, $L^\perp$ tiene 1 palabra de peso 0 y 3 palabras de peso 2. De esta manera obtenemos que $L^\perp = \{000, 011, 101, 110\}$. □

Existe una definición alternativa del enumerador de peso, que a veces resulta muy útil. Si L es un $[n, k]$-código lineal, el *enumerador de peso homogéneo* es el polinomio

$$W_L(x, y) = \sum_{i=0}^{n} A_i x^{n-i} y^i.$$

Se puede ver que, en este contexto, la identidad de MacWilliams toma la forma

$$W_{L^\perp}(x, y) = \tfrac{1}{|L|} W_L\big(x + (q-1)y, x - y\big).$$

El siguiente resultado, que no probaremos, es muy útil y lo usaremos más adelante.

Teorema 2.29. *Sea L un $[n, n/2]$-código autodual q-ario. Entonces, su enumerador de peso es un polinomio en $p(x, y) = y(x - y)$ y en $q(x, y) = x^2 + (q-1)y^2$, es decir*

$$W_L(x, y) = \sum_{\substack{i,j \\ 2(i+j)=n}} c_{ij}\, p(x, y)^i \, q(x, y)^j.$$

Si además, el peso de cualquier palabra código es múltiplo de 4, entonces

$$W_L(x, y) = \sum_{\substack{i,j \\ 8i+24j=n}} c_{ij}\, r(x, y)^i \, s(x, y)^j, \tag{2.7}$$

donde $r(x, y) = x^8 + 14x^4y^4 + y^8$ y $s(x, y) = x^4y^4(x^4 - y^4)^4$.

3. Algunos Códigos Lineales Famosos

En esta sección introducimos los códigos de Hamming, de Golay y de Reed-Muller. Para los dos primeros mostraremos como realizar el proceso de decodificación.

3.1. Códigos de Hamming. Comencemos con un ejemplo. Consideremos la matriz

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \in M_{3\times 7}(\mathbb{F}_2) \tag{3.1}$$

cuyas columnas son las $2^3 - 1 = 7$ palabras no-nulas de $\mathbb{F}_2^3$, escritas en forma ascendente, es decir la representación binaria ordenada de los números del 1 al 7. Pensemos que H es la matriz de paridad de un código lineal C con parámetros $[7, 4, d]$. Podemos calcular la distancia mínima de C usando el Teorema 2.14. Como cualquier par de columnas de H son linealmente independientes, pero H tiene 3 columnas linealmente dependientes, entonces C tiene distancia $d = 3$. Para encontrar las $2^4 = 16$ palabras código de C sólo tenemos que resolver las ecuaciones de paridad determinadas por H

$$\begin{cases} & & & x_4 & +x_5 & +x_6 & +x_7 & = 0 \\ & x_2 & +x_3 & & & +x_6 & +x_7 & = 0 \\ x_1 & & +x_3 & & +x_5 & & +x_7 & = 0 \end{cases}$$

Una base de C se obtiene tomando a x_3, x_5, x_6 y x_7 como variables libres. Es decir $c_1 = 1110000$, $c_2 = 1101100$, $c_3 = 0101010$ y $c_4 = 11010001$. Luego, el código C se obtiene haciendo todas las sumas posibles entre c_1, c_2, c_3 y c_4. Este es el código de Hamming binario de longitud 7, denotado por $\mathcal{H}_2(3)$.

La construcción anterior se puede hacer para cualquier $r \geq 2$. Es decir, si formamos la matriz H cuyas columnas son las $2^r - 1$ palabras no-nulas de $\mathbb{F}_2^r$, tenemos una matriz $r \times n$, con $n = 2^r - 1$, en donde cualquier par de columnas son linealmente independientes, pero hay 3 columnas linealmente dependientes. Luego, H es la matriz de paridad de un código lineal denotado por $\mathcal{H}_2(r)$ con parámetros

$$n = 2^r - 1, \qquad k = n - r = 2^r - r - 1, \qquad d = 3.$$

Este es el llamado *código de Hamming binario de orden* r. Por ejemplo, $\mathcal{H}_2(4)$ tiene parámetros $[15, 11, 3]$, luego codifica $2^{11} = 2048$ mensajes y corrige 1 error.

Utilizando el procedimiento anterior, los códigos de Hamming binarios pueden generalizarse a cualquier alfabeto $\mathbb{F}_q$. Para cada r, queremos construir una matriz $H_{q,r} \in M_{r\times n}(\mathbb{F}_q)$, con el mayor número de columnas, de modo que cualquier par de columnas sean linealmente independientes (o sea ninguna columna es múltiplo de otra), pero que algún conjunto de tres columnas sea linealmente dependiente.

Para cada r fijo, construimos la matriz $H_{q,r}$ de la siguiente manera. Elegimos cualquier columna no-nula $c_1 \in V_1 = \mathbb{F}_q^r$. Luego elegimos cualquier columna no-nula

$$c_2 \in V_2 = V_1 \setminus \{\alpha c_1 : \alpha \in \mathbb{F}_q^*\}.$$

Continuamos eligiendo columnas no-nulas de esta forma y descartamos los múltiplos escalares de las columnas elegidas hasta agotar todas las columnas de $\mathbb{F}_q^r$. Como cada columna $c \in \mathbb{F}_q^r$ tiene $q - 1$ múltiplos escalares no-nulos αc, $\alpha \in \mathbb{F}_q$, vemos que la matriz $H_{q,r}$ formada por las columnas c_i elegidas como antes tiene $(q^r - 1)/(q - 1)$ columnas.

Definición 3.1. La matriz $H_{q,r} \in M_{r\times n}(\mathbb{F}_q)$, con $n = (q^r-1)/(q-1)$, se llama *matriz de Hamming de orden r* y es la matriz de paridad de un código lineal q-ario con parámetros

$$n = \frac{q^r-1}{q-1}, \qquad k = n-r, \qquad d = 3,$$

denotado por $\mathcal{H}_q(r)$ y llamado *código de Hamming q-ario de orden r*.

Los códigos de Hamming $\mathcal{H}_q(r)$ fueron descubiertos independientemente por Marcel Golay (1949) y por Richard Hamming (1950). Sus parámetros $(\frac{q^r-1}{q-1}, q^{n-r}, 3)$ satisfacen la igualdad en la cota de Hamming (1.7). En efecto,

$$q^k \sum_{i=0}^{1} \tbinom{n}{i}(q-1)^i = q^k(1+n(q-1)) = q^{k+r} = q^n.$$

Por lo tanto, los códigos de Hamming $\mathcal{H}_q(r)$ son códigos perfectos, y además corrigen 1 error.

Una forma fácil de construir una matriz $H_{q,r}$ es tomar todos los vectores en $\mathbb{F}_q^r$ cuya primer coordenada no-nula es 1 (notar que en el caso $q = 2$, esto es equivalente a tomar todas las palabras en orden ascendente). En efecto, hay $q^r - 1$ vectores no-nulos y el primer elemento no-nulo puede ser $1, 2, \ldots, q-1$. Luego, hay $(q^r-1)/(q-1)$ vectores cuya primer coordenada no-nula comienza con 1.

Ejemplo 3.2. El código $\mathcal{H}_3(3)$ tiene parámetros $[13, 10, 3]$ con matriz de paridad

$$H_{3,3} = \begin{pmatrix} 0&0&0&0&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&0&0&0&1&1&1&2&2&2 \\ 1&0&1&2&0&1&2&0&1&2&0&1&2 \end{pmatrix}.$$

Ejemplo 3.3. El código $\mathcal{H}_5(3)$ tiene parámetros $[31, 26, 3]$ con matriz de paridad

$$H_{3,5} = \begin{pmatrix} 0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\ 0&1&1&1&1&1&0&0&0&0&0&1&1&1&1&1&2&2&2&2&2&3&3&3&3&3&4&4&4&4&4 \\ 1&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4&0&1&2&3&4 \end{pmatrix}.$$

Observación 3.4. Notar que la elección de las columnas en la construcción previa no es única y por lo tanto hay muchos códigos de Hamming con los mismos parámetros. Sin embargo, todos los códigos de Hamming de igual tamaño son múltiplo escalar equivalentes. Más aún, todo código lineal con los parámetros de Hamming es múltiplo escalar equivalente a un código de Hamming.

Observación 3.5. Hay códigos no-lineales con los parámetros de Hamming. Luego, hay códigos perfectos no-lineales. La siguiente construcción se debe a Vasil'ev (1962). Sea $\eta : \mathcal{H}_2(r) \to \mathbb{Z}_2$ un mapa no lineal con $\eta(0) = 0$. Luego, existen $c_1, c_2 \in \mathcal{H}_2(r)$ tales que $\eta(c_1 + c_2) \neq \eta(c_1) + \eta(c_2)$. Para cada $x \in \mathbb{Z}_2^n$, sea $\varepsilon(x) = 0$, si x tiene peso par, y $\varepsilon(x) = 1$, si x tiene peso impar. Consideremos el código

$$\mathcal{V} = \{(x, x+c, \varepsilon(x) + \eta(c)) : x \in \mathbb{Z}_2^n, c \in \mathcal{H}_2(r)\},$$

donde $n = 2^r - 1$. Dejamos como ejercicio chequear que $\mathcal{V}$ es un código binario no-lineal perfecto, con los parámetros de Hamming $(2^{r+1} - 1, 2^{2n-r}, 3)$.

La decodificación por síndrome con algunas matrices de Hamming es más elegante que con otras. Las matrices descriptas más arriba (primera coordenada no-nula igual a 1) son un ejemplo de esto.

En el caso binario, si cometemos un error en la transmisión en la i-ésima coordenada, el vector error que resulta es e_i. Luego, el síndrome de la palabra recibida es $e_i H^\intercal$, que es la transpuesta de la i-ésima columna H_i de H. Más aún, el número binario que representa este vector coincide con la posición del error, es decir i.

Ejemplo 3.6. Sea $\mathcal{H}_2(3)$ el código de Hamming definido por la matriz $H_{3,2} = H$ de (3.1). Supongamos que cometemos un error en la tercer coordenada, es decir el patrón de error es $e_3 = 0010000$. Luego, $e_3 H^\intercal = (0010000)H^\intercal = (H_3)^\intercal = (011) = (011)_2 = 3_{10}$. Es decir, ¡el síndrome determina la coordenada errónea! □

El caso general, para $\mathcal{H}_q(r)$, es muy parecido. Si cometemos un error en la coordenada i, el vector error es de la forma αe_i, con $\alpha \in \mathbb{F}_q^*$. Luego, el síndrome es $s(e_i) = \alpha e_i H_{q,r}^\intercal$, que es α multiplicado por la i-ésima columna de $H_{q,r}$ (transpuesta). Por la forma en que construimos H, vemos que α es la primera coordenada no-nula de $s(e_i)$. Multiplicando $s(e_i)$ por α^{-1} obtenemos la columna i de H, lo cual nos da la coordenada del error.

Ejemplo 3.7. Sea $\mathcal{H}_3(3)$ el código de Hamming definido por la matriz $H = H_{3,3}$ del Ejemplo 3.2. Si recibimos la palabra $x = 1101112211201$, su síndrome es

$$xH^\intercal = (201) = 2(102) = 2 \times (\text{columna 7 de } H).$$

Por lo tanto, detectamos que hay un error de magnitud 2 en la coordenada 7 de la palabra recibida x. Luego, decodificamos x como

$$c = x - 2e_7 = 1101112211201 - 00000020000 = 1101110211201.$$

3.2. Códigos de Golay. Ahora presentamos los 4 códigos $\mathcal{G}_{24}$, $\mathcal{G}_{23}$, $\mathcal{G}_{12}$ y $\mathcal{G}_{11}$ introducidos por Marcel Golay en 1949.

El *código de Golay* $\mathcal{G}_{24}$ es el código lineal binario definido por la matriz generadora

$$G = (Id_{12} \,|\, A) \in M_{12\times 24}(\mathbb{F}_2), \tag{3.2}$$

donde

$$A = \begin{pmatrix} 0&1&1&1&1&1&1&1&1&1&1&1\\ 1&1&1&0&1&1&1&0&0&0&1&0\\ 1&1&0&1&1&1&0&0&0&1&0&1\\ 1&0&1&1&1&0&0&0&1&0&1&1\\ 1&1&1&1&0&0&0&1&0&1&1&0\\ 1&1&1&0&0&0&1&0&1&1&0&1\\ 1&1&0&0&0&1&0&1&1&0&1&1\\ 1&0&0&0&1&0&1&1&0&1&1&1\\ 1&0&0&1&0&1&1&0&1&1&1&0\\ 1&0&1&0&1&1&0&1&1&1&0&0\\ 1&1&0&1&1&0&1&1&1&0&0&0\\ 1&0&1&1&0&1&1&1&0&0&0&1 \end{pmatrix}. \tag{3.3}$$

Notar la estructura cíclica que posee A. Veremos que $\mathcal{G}_{24}$ tiene parámetros $[24, 12, 8]$ y, por lo tanto, corrige 3 errores. Sólo falta ver que $d = 8$.

Teorema 3.8. *El código de Golay $\mathcal{G}_{24}$ tiene las siguientes propiedades.*

(i) *$\mathcal{G}_{24}$ es autodual, es decir $\mathcal{G}_{24}^\perp = \mathcal{G}_{24}$.*

(ii) *$\mathcal{G}_{24}$ está generado por la matriz $G = (A \,|\, Id_{12})$.*

(iii) *Si $c \in \mathcal{G}_{24}$ entonces $4 \,|\, w(c)$.*

(iv) *$\mathcal{G}_{24}$ no tiene palabras de peso 4.*

(v) *$\mathcal{G}_{24}$ es un $[24, 12, 8]$-código.*

Demostración. Sea G la matriz generadora de $\mathcal{G}_{24}$ dada en (3.2) y (3.3).

(i) Como las filas de G son ortogonales, entonces todo par de palabras código en $\mathcal{G}_{24}$ son ortogonales. Luego, $\mathcal{G}_{24} \subset \mathcal{G}_{24}^\perp$. Pero como $\dim \mathcal{G}_{24}^\perp = \dim \mathcal{G}_{24}$, entonces vale $\mathcal{G}_{24}^\perp = \mathcal{G}_{24}$.

(ii) Sigue de (i) y del hecho que $A^\intercal = A$.

(iii) El peso de las filas de G es 8 ó 12, por lo tanto divisible por 4. Si x e y son filas de G entonces, por (1.4), tenemos

$$w(x+y) = w(x) + w(y) - 2w(x \cap y).$$

Pero $w(x \cap y) \equiv x \cdot y = 0 \mod (2)$ y por lo tanto $x+y$ tiene peso par. Por inducción, el peso de la suma de cualquier número de filas de G es múltiplo de 4.

(iv) Usaremos las dos matrices generadoras $G_1 = (Id_{12} \mid A)$ y $G_2 = (A \mid Id_{12})$ de $\mathcal{G}_{24}$. Si $c \in \mathcal{G}_{24}$ escribimos $c = c_L c_R$ donde $c_L, c_R \in \mathbb{F}_2^{12}$ son la parte izquierda y derecha de c, respectivamente. Supongamos que $w(c) = 4$. Notar que cualquier combinación lineal de las filas de G_1 tiene parte izquierda con peso mayor que 1. Usando G_2, la parte derecha tiene peso mayor que 1. Entonces $w(c_L) \geq 1$ y $w(c_R) \geq 1$. Ahora, si $w(c_L) = 1$, entonces c es una fila de G y por lo tanto $w(c) \neq 4$. Luego, $w(c_L) \geq 2$ y, análogamente $w(c_R) \geq 2$. Por lo tanto, la única posibilidad es $w(c_L) = w(c_R) = 2$. Luego, c es la suma de dos filas x e y de G_1. Absurdo, pues $w(x+y) \neq 4$ para todo par de filas x, y de G_1. Luego $\mathcal{G}_{24}$ no tiene palabras de peso 4.

(v) Por (iii) y (iv) tenemos $w_{\mathcal{G}_{24}} \geq 8$, y la fila 2 de G tiene peso 8, luego $d_{\mathcal{G}_{24}} = 8$. □

Pregunta 3.4. ¿Cuál es la distribución de los pesos en $\mathcal{G}_{24}$? Es decir, ¿Cuántas palabras de un peso dado hay en $\mathcal{G}_{24}$?

El código $\mathcal{G}_{24}$ tiene $2^{12} = 4096$ palabras código, por lo que el método de inspección no parece el más adecuado. Para responder esta pregunta usaremos los *enumeradores de peso* para códigos autoduales vistos en la Sección 2.5. Recordemos que, dado un código C, A_k denota el número de palabras de peso k en C. Queremos encontrar la distribución de pesos $A_0, A_1, \ldots, A_{24}$ de $\mathcal{G}_{24}$. Por el Teorema 3.8, tenemos que los posibles A_i's no-nulos son $A_0, A_8, A_{12}, A_{16}, A_{20}$ y A_{24}. Como $\mathbf{1} \in \mathcal{G}_{24}$ (¿porqué?) y $\mathcal{G}_{24}$ es autodual, deducimos que

$$A_0 = A_{24} = 1, \qquad A_4 = A_{20} = 0, \qquad A_8 = A_{16} = ?, \qquad A_{12} = ?$$

Usando esta información, el enumerador de peso homogéneo de $\mathcal{G}_{24}$ se simplifica bastante y, por definición, tenemos que

$$W_{\mathcal{G}_{24}}(x,y) = x^{24} + A_8 x^{16} y^8 + A_{12} x^{12} y^{12} + A_{16} x^8 y^{16} + y^{24}. \tag{3.5}$$

Ahora, como el peso de toda palabra en $\mathcal{G}_{24}$ es múltiplo de 4, por (2.7) del Teorema 2.29, también se tiene

$$W_{\mathcal{G}_{24}}(x,y) = \sum_{\substack{i,j \\ 8i+24j=24}} c_{ij}\, r(x,y)^i\, s(x,y)^j \tag{3.6}$$

con

$$r(x,y) = x^8 + 14x^4y^4 + y^8 \qquad \text{y} \qquad s(x,y) = x^4y^4(x^4 - y^4)^4.$$

Pero

$$24 = 8i + 24j \qquad \text{si y sólo si} \qquad (i,j) = (3,0) \ \text{ó} \ (i,j) = (0,1),$$

luego, (3.6) queda

$$W_{\mathcal{G}_{24}}(x,y) = a\, r(x,y)^3 + b\, s(x,y). \tag{3.7}$$

Ahora, igualando los coeficientes de (3.5) y (3.7) se ve que $a = 1$ y $b = -42$. Usando esto, se obtiene $A_8 = 759$ y $A_{12} = 2576$. Si el lector desconfía, con todo derecho, de los números obtenidos, puede realizar los detalles de los cálculos a modo de ejercicio. Como un buen indicio, tenemos $2 + 2 \cdot 759 + 2576 = 4096 = 2^{12}$.

Finalmente, la distribución de pesos de $\mathcal{G}_{24}$ es

$$A_0 = A_{24} = 1, \qquad A_4 = A_{20} = 0, \qquad A_8 = A_{16} = 759, \qquad A_{12} = 2576.$$

Pinchando el código de Golay en la última coordenada se obtiene el *código de Golay pinchado* $\mathcal{G}_{23} = \mathcal{G}_{24}^*$, con parámetros $[23, 12, 7]$. Por lo tanto, este código es perfecto. Se puede ver que pinchando el código $\mathcal{G}_{24}$ en cualquier coordenada se obtienen códigos equivalentes. Al revés, podemos recuperar $\mathcal{G}_{24}$ extendiendo el código $\mathcal{G}_{23}$ con un dígito extra de paridad.

El código *ternario de Golay* $\mathcal{G}_{12}$ es el código definido por la matriz generadora

$$G = (Id_6 \,|\, B) \in M_{6\times 12}(\mathbb{F}_3), \tag{3.8}$$

donde

$$B = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 & 2 \\ 1 & 2 & 1 & 0 & 1 & 2 \\ 1 & 2 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}. \tag{3.9}$$

Nuevamente, notemos la naturaleza cíclica de la matriz B.

Al igual que con los códigos binarios de Golay, pinchando el código $\mathcal{G}_{12}$ en cualquier coordenada se obtienen códigos equivalentes, y agregando un dígito extra de paridad a cualquiera de éstos se obtiene $\mathcal{G}_{12}$.

Dejamos como ejercicio para el lector, el siguiente resultado que reune las propiedades de los códigos ternarios de Golay.

Teorema 3.9. *El código ternario de Golay $\mathcal{G}_{12}$ tiene las siguientes propiedades.*

(i) *$\mathcal{G}_{12}$ es autodual, es decir $\mathcal{G}_{12}^{\perp} = \mathcal{G}_{12}$.*

(ii) *B es simétrica y $\mathcal{G}_{12}$ está generado por la matriz $G = (B \,|\, Id_6)$.*

(iii) *$\mathcal{G}_{12}$ es un $[12, 6, 6]_3$-código.*

(iv) *El código $\mathcal{G}_{11}$ obtenido pinchando $\mathcal{G}_{12}$ en la última coordenada tiene parámetros $[11, 6, 5]$ y por lo tanto es perfecto.*

Observación 3.10. Los códigos de Golay son únicos. Es decir, todo código (lineal o no) con los parámetros de un código de Golay es equivalente a un código de Golay con dichos parámetros. En efecto, todo código C con parámetros $(24, 2^{12}, 8)_2$, $(23, 2^{12}, 7)_2$, $(12, 3^6, 6)_3$ ó $(11, 3^6, 5)_3$ es equivalente a $\mathcal{G}_{24}$, $\mathcal{G}_{23}$, $\mathcal{G}_{12}$ ó $\mathcal{G}_{11}$, respectivamente. Esto fue probado por Pless (1968) para códigos lineales y por Delsarte y Goethals (1975) en el caso general, usando el resultado previo de Pless.

Decodificación de $\mathcal{G}_{24}$. Como $\mathcal{G}_{24}$ es un código con parámetros $[24, 12, 8]$, la decodificación por síndrome requeriría construir $\frac{2^{24}}{2^{12}} = 2^{12} = 4096$ síndromes. Como $\mathcal{G}_{24}$ es 3-corrector, podríamos usar la tabla acortada, sólo con los líderes de peso ≤ 3 o menos (sabemos que las palabras de peso 3 o menos son líderes). Aún así, tendríamos que considerar

$$1 + 24 + \tbinom{24}{2} + \tbinom{24}{3} = 2325$$

síndromes.

Veamos una estrategia de decodificación, que saca provecho de la estructura de $\mathcal{G}_{24}$ y de sus propiedades, para corregir todos los errores de peso 3 o menos (que es lo máximo que este código puede corregir). Como $\mathcal{G}_{24}$ es autodual, las matrices generadoras $G_1 = (Id \,|\, A)$ y $G_2 = (A \,|\, Id)$ son también matrices de paridad. Supongamos que x es la palabra recibida y e es el error, con $w(e) \leq 3$. Escribimos

$$e = e_1 e_2,$$

donde e_1 y e_2 tienen longitud 12. Calculando los síndromes de dos maneras distintas, tenemos

$$s_1 = s_{G_1}(e) = eG_1^\intercal = (e_1 \mid e_2)\binom{Id}{A} = e_1 + e_2A,$$

y similarmente,

$$s_2 = s_{G_2}(e) = eG_2^\intercal = (e_1 \mid e_2)\binom{A}{Id} = e_1A + e_2.$$

Las posibilidades son:

1) Si $w(e_1) = 0$, entonces $e = 0s_2$ y $w(s_2) \leq 3$.
2) Si $w(e_2) = 0$, entonces $e = s_10$ y $w(s_1) \leq 3$.
3) Si $w(e_1) > 0$ y $w(e_2) > 0$, entonces $w(s_1) \geq 5$ y $w(s_2) \geq 5$.

Luego, si algún síndrome tiene peso a lo sumo 3, recuperamos el vector error e. Por otra parte, si $w(s_1) > 3$ y $w(s_1) > 3$ entonces hay dos posibilidades:

3a) $w(e_1) = 1$ y $1 \leq w(e_2) \leq 2$.
3b) $w(e_1) = 2$ y $w(e_2) = 1$.

Caso 3a): Sean ϵ_j los vectores de la base canónica de $\mathbb{Z}_2^{12}$. Si cometemos un error en las primeras 12 coordenadas, digamos en la coordenada i, entonces $e_1 = \epsilon_i$ y $e = \epsilon_i e_2$. Para $j = 1, \ldots, 12$ calculamos

$$(x + \epsilon_j 0)G_2^\intercal = (e + \epsilon_j 0)G_2^\intercal = (\epsilon_i e_2 + \epsilon_j 0)G_2^\intercal = \epsilon_i A + e_2 + \epsilon_j A.$$

Ahora, si $j = i$, el vector $(x + \epsilon_j 0)G_2^\intercal = e_2$ tiene peso a lo sumo 2, mientras que si $j \neq i$, tiene peso por lo menos 4. Entonces, chequeando los 12 síndromes

$$(x + \epsilon_1 0)G_2^\intercal, \ldots, (x + \epsilon_{12} 0)G_2^\intercal,$$

podemos determinar la coordenada i en donde está el error en e_1.

Caso 3b): Si cometemos un error en la coordenada i, en las últimas 12 coordenadas, entonces $e_2 = \epsilon_i$ y $e = e_1\epsilon_i$. En este caso, calculamos

$$(x + 0\epsilon_j)G_1^\intercal = (e + 0\epsilon_j)G_1^\intercal = (e_1\epsilon_i + 0\epsilon_j)G_1^\intercal = e_1 + \epsilon_i A + \epsilon_j A.$$

Como antes, $(x + 0\epsilon_j)G_1^\intercal$ tiene peso a lo sumo 2, si $j = i$, y tiene peso por lo menos 4, si $j \neq i$. Luego, determinamos la coordenada i en donde está el error en e_2.

Se concluye que, si al transmitir una palabra se cometieron a lo sumo 3 errores, podemos decodificarla correctamente calculando los síndromes

$$xG_1^\intercal, xG_2^\intercal, (x + \epsilon_1 0)G_2^\intercal, \ldots, (x + \epsilon_{12} 0)G_2^\intercal, (x + 0\epsilon_1)G_1^\intercal, \ldots, (x + 0\epsilon_{12})G_1^\intercal.$$

Es decir, sólo se necesitan calcular ¡26 síndromes!

3.3. Códigos de Reed-Muller. Veamos ahora una familia de códigos lineales muy usados en la práctica, a pesar de ser uno de los más viejos. Estos códigos fueron estudiados por primera vez por I. E. Reed (1954) y por D. E. Muller (1954). Hay uno de estos códigos para cada par $r, m \in \mathbb{N}$, con $0 \leq r \leq m$. Existen, sin embargo, muchas formas alternativas de definirlos. Por ejemplo, a través de polinomios booleanos en m variables de grado a lo sumo r, ó, como funciones características de hiperplanos afines de la geometría euclídea finita $EG(m, 2)$. No entraremos en estos detalles y daremos una definición recursiva de estos códigos, utilizando la construcción $(u, u + v)$ dada en la Sección 1.8.

Definición 3.11. Para $0 \leq r \leq m$, el *código de Reed-Muller* de orden r y longitud 2^m, denotado por $\mathcal{R}(r, m)$, se define de la siguiente manera:

1. $\mathcal{R}(0, m) = Rep_2(2m) = \{\mathbf{0}, \mathbf{1}\}$,
2. $\mathcal{R}(m, m) = \mathbb{Z}_2^{2^m}$,
3. $\mathcal{R}(r, m) = \mathcal{R}(r, m-1) \oplus \mathcal{R}(r-1, m-1)$, si $0 < r < m$.

Es decir, si $0 < r < m$, tenemos

$$\mathcal{R}(r,m) = \{(u, u+v) \in \mathbb{Z}_2^{2^m} : u \in \mathcal{R}(r, m-1), v \in \mathcal{R}(r-1, m-1)\}.$$

Ejemplo 3.12. Veamos estos códigos para $m \leq 3$. Los primeros casos son triviales. En efecto, $\mathcal{R}(0,0) = \{0,1\}$, $\mathcal{R}(0,1) = \{00,11\}$ y $\mathcal{R}(1,1) = \{00,01,10,11\}$. Además, tenemos $\mathcal{R}(0,2) = \{0000,1111\}$ y $\mathcal{R}(2,2) = \mathbb{Z}_2^4$. El primer caso no trivial es $r = 1$ y $m = 2$. Por definición,

$$\mathcal{R}(1,2) \;=\; \{(u, u+v) \in \mathbb{Z}_2^4 : u \in \mathcal{R}(1,1), v \in \mathcal{R}(0,1)\}.$$

Luego, las palabras códigos son

$$\begin{array}{llll} (00, 00+00), & (01, 01+00), & (10, 10+00), & (11, 11+00), \\ (00, 00+11), & (01, 01+11), & (10, 10+11), & (11, 11+11), \end{array}$$

y por lo tanto se tiene

$$\mathcal{R}(1,2) = \{0000, 0011, 0101, 0110, 1010, 1001, 1111, 1100\} = E(4).$$

Si $m = 3$, además de los triviales $\mathcal{R}(0,3) = Rep_2(8)$ y $\mathcal{R}(3,3) = \mathbb{Z}_2^8$, tenemos

$$\mathcal{R}(1,3) = \mathcal{R}(1,2) \oplus \mathcal{R}(0,2),$$

$$\mathcal{R}(2,3) = \mathcal{R}(2,2) \oplus \mathcal{R}(1,2).$$

Dejamos como ejercicio calcular estos códigos. □

Damos a continuación, sin demostración, las propiedades más importantes de los códigos de Reed-Muller.

Teorema 3.13. (1) *$\mathcal{R}(r,m)$ es un código lineal binario con parámetros*

$$n = 2^m, \qquad k = \sum_{i=0}^{r} \tbinom{m}{i}, \qquad d = 2^{m-r}.$$

(2) *$\mathcal{R}(r,m)$ tiene matriz generadora*

$$G_{r,m} = \begin{pmatrix} G_{r,m-1} & G_{r,m-1} \\ 0 & G_{r-1,m-1} \end{pmatrix}, \qquad \textit{si } 0 < r < m,$$

$$G_{0,m} = \underbrace{(11\cdots1)}_{(m+1)-veces} \qquad \textit{ó} \qquad G_{m,m} = \begin{pmatrix} G_{m-1,m} \\ 0\cdots01 \end{pmatrix}.$$

(3) *$\mathcal{R}(r-1,m) \subset \mathcal{R}(r,m)$ para $r > 0$. Es decir,*

$$\mathcal{R}(0,m) \subset \mathcal{R}(1,m) \subset \cdots \subset \mathcal{R}(r-1,m) \subset \mathcal{R}(r,m).$$

(4) *$\mathcal{R}(m-1,m) = E(2^m)$. Luego, si $r < m$, $\mathcal{R}(r,m)$ contiene sólo palabras de peso par.*

(5) *Todas las palabras en $\mathcal{R}(1,m)$, salvo $\mathbf{0}$ y $\mathbf{1}$, tienen peso 2^{m-1}.*

(6) *$\mathcal{R}^{\perp}(r,m) = \mathcal{R}(m-1-r,m)$, con $r < m$. Luego, $\mathcal{R}(r, 2r+1)$ es autodual.*

4. Códigos Cíclicos

4.1. Polinomio generador. Supondremos que n y q son coprimos. En particular, si $q = 2$ entonces n es impar.

Definición 4.1. Un código lineal $C \subset \mathbb{F}_q^n$ es *cíclico* si

$$c_0c_1 \dots c_{n-1} \in C \quad \Rightarrow \quad c_{n-1}c_0 \dots c_{n-2} \in C.$$

Notar que, por definición, un código lineal C es cíclico si es cerrado por el *desplazamiento cíclico* $c_0c_1 \dots c_{n-1} \mapsto c_{n-1}c_0 \dots c_{n-2}$. En este caso, C es cerrado por todos los desplazamientos cíclicos $c_0c_1 \dots c_{n-1} \mapsto c_k \dots c_{n-1}c_0 \dots c_{k-1}$.

Si $C \subset \mathbb{F}_q^n$ es un código lineal q-ario, a cada palabra código podemos asignarle un polinomio como sigue:

$$\phi : C \to \mathbb{F}_q[x], \qquad c_0c_1 \dots c_{n-1} \mapsto c_0 + c_1x + \cdots + c_{n-1}x^{n-1}.$$

El mapa ϕ es un isomorfismo de espacio vectorial de C sobre $\phi(C)$. Luego, de ahora en adelante, ignoraremos el mapa ϕ y pensaremos a las palabras códigos como polinomios, y recíprocamente.

El cociente

$$R_n = \frac{\mathbb{F}_q[x]}{\langle x^n - 1 \rangle}$$

es el álgebra de polinomios de grado menor que n, con la suma usual de polinomios y el producto de polinomios seguido de reducción módulo $x^n - 1$.

Observación 4.2. Un código C es cíclico si y sólo si $\phi(C)$ es un ideal en $\mathbb{F}_q[x]/\langle x^n - 1\rangle$. En efecto, si $c_0c_1 \dots c_{n-1} \in C$, entonces

$$\begin{aligned} x(c_0 + c_1x + \cdots + c_{n-1}x^{n-1}) &= c_0x + c_1x^2 + \cdots + c_{n-1}x^n \mod (x^n - 1) \\ &= c_{n-1} + c_0x + \cdots + c_{n-2}x^{n-1}. \end{aligned}$$

El siguiente resultado reune algunos hechos básicos de los códigos cíclicos.

Teorema 4.3. *Sea C un ideal de R_n, es decir un código cíclico de longitud n. Entonces:*

(1) *Existe un único polinomio mónico $g(x)$ de grado mínimo en C. Además, este polinomio genera C, es decir $C = \langle g(x)\rangle$.*

(2) $g(x) \,|\, x^n - 1$.

(3) *Si* $\operatorname{gr}(g(x)) = r$, *entonces C tiene dimensión $n - r$. Más aún,*

$$C = \langle g(x)\rangle = \{r(x)g(x) : \operatorname{gr}(g(x)) < n - r\}.$$

(4) *Si $g(x) = g_0 + g_1x + \cdots + g_rx^r$, entonces $g_0 \neq 0$ y C tiene matriz generadora*

$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & \cdots & g_r & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & \cdots & g_r & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \cdots & \cdots & g_r & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & & & \ddots & 0 \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & \cdots & g_r \end{pmatrix}$$

donde cada fila de G es un desplazamiento cíclico de la fila previa.

Demostración. (1) Supongamos que C contiene 2 polinomios mónicos distintos, $g_1(x)$ y $g_2(x)$, de grado mínimo r. Entonces, $g_1(x) - g_2(x)$ es un polinomio no-nulo de grado menor que r, lo cual es absurdo. Luego, existe un único polinomio mónico de grado mínimo r en C.

Como $g(x) \in C$, y C es un ideal, tenemos que $\langle g(x)\rangle \subset C$. Por otra parte, supongamos que $p(x) \in C$. Existen $q(x), r(x)$ tales que

$$p(x) = q(x)g(x) + r(x), \qquad 0 \leq \operatorname{gr}(r(x)) < r.$$

Luego, como $r(x) = p(x) - q(x)g(x) \in C$, y tiene grado menor que r, necesariamente $r(x) = 0$. Así, $p(x) = q(x)g(x) \in \langle g(x)\rangle$ y $C \subset \langle g(x)\rangle$. Por lo tanto, $\langle g(x)\rangle = C$.

(2) Dividiendo $x^n - 1$ por $g(x)$ tenemos

$$x^n - 1 = q(x)g(x) + r(x)$$

y $0 \leq \operatorname{gr}(r(x)) < r$. Como en R_n se tiene que $x^n - 1 = 0 \in C$, vemos que $r(x) \in C$ y por lo tanto $r(x) = 0$.

(3) El ideal generado por $g(x)$ es $\langle g(x)\rangle = \{f(x)g(x) : f(x) \in R_n\}$. Queremos ver que basta restringir $f(x)$ a polinomios de grado menor que $n - r$. Sabemos que $x^n - 1 = h(x)g(x)$ para algún polinomio $h(x)$ de grado $n - r$. Dividiendo, $f(x) = q(x)h(x) + r(x)$ con $\operatorname{gr}(r(x)) < n - r$. Entonces,

$$f(x)g(x) = q(x)h(x)g(x) + r(x)g(x) = q(x)(x^n - 1) + r(x)g(x),$$

y así $f(x)g(x) = r(x)g(x)$ en R_n, que es lo que queríamos ver. Esto también muestra que el conjunto

$$\{g(x), xg(x), \ldots, x^{n-r+1}g(x)\}$$

genera C, y como es linealmente independiente, forma una base de C. Luego, $\dim C = n - r$.

(4) Si $g_0 = 0$, entonces $g(x) = xg_1(x)$ con $\operatorname{gr}(g_1(x)) < r$. Pero entonces tenemos

$$g_1(x) = 1 \cdot g_1(x) \equiv x^n g_1(x) = x^{n-1}g(x) \in C,$$

lo cual es absurdo pues $g_1 \neq 0$ tiene grado menor que $g(x)$. Por lo tanto $g_0 = 0$. Por último, G es matriz generadora de C, pues $\{g(x), xg(x), \ldots, x^{n-r+1}g(x)\}$ es una base de C. □

Es importante notar que el código cíclico C puede estar generado por otros polinomios además del polinomio generador, como lo muestra el siguiente ejemplo.

Ejemplo 4.4. Como $1+x$ divide a x^3-1, $C = \langle 1+x\rangle$ es un código cíclico en $R_3 = \mathbb{F}_2[x]/\langle x^3-1\rangle$. Por el teorema anterior, $\dim C = 3 - 1 = 2$ y C está formado por los múltiplos de $1 + x$:

$$0, \qquad 1 + x, \qquad x(1 + x) = x + x^2, \qquad (1 + x)(1 + x) = 1 + x^2.$$

Luego,

$$C = \{0, 1 + x, 1 + x^2, x + x^2\} = \{000, 110, 101, 011\} = E(3).$$

Notar que

$$\langle 1 + x^2\rangle = \{0, 1 + x^2, x(1 + x^2), (1 + x)(1 + x^2)\} = \{0, 1 + x^2, 1 + x, x + x^2\} = C.$$

Es decir, C también está generado por $1 + x^2$. Notar que $1 + x^2$ no divide a $x^3 - 1$. □

En el ejemplo anterior vimos que $C = \langle 1 + x\rangle = \langle 1 + x^2\rangle$. Cuando haya peligro de confusión, adoptaremos la notación $C = \langle\langle p(x)\rangle\rangle$ para denotar el hecho que C es el ideal generado por $p(x)$ y que $p(x)$ es el polinomio generador de C.

Una condición necesaria para que $g(x)$ sea el polinomio generador de un código cíclico de longitud n es que divida a $x^n - 1$. Veamos que ésta es también una condición suficiente.

Proposición 4.5. *Un polinomio mónico $p(x)$ en R_n es el polinomio generador de un código cíclico de longitud n si y sólo si $p(x) \mid x^n - 1$.*

Demostración. Supongamos que $p(x) \mid x^n-1$ y que $g(x)$ es el polinomio generador de $C = \langle p(x)\rangle$, con $p(x) \neq g(x)$. Como $p(x)$ y $g(x)$ son mónicos, entonces $\operatorname{gr}(g(x)) < \operatorname{gr}(p(x))$.

Por hipótesis,

$$x^n - 1 = p(x)f(x)$$

para algún polinomio $f(x) \neq 0$. Más aún, como $g(x) \in \langle p(x)\rangle$, entonces

$$g(x) \equiv a(x)p(x)$$

para algún $a(x) \in R_n$. Luego, tenemos

$$g(x)f(x) \equiv a(x)p(x)f(x) \equiv a(x)(x^n - 1) \equiv 0.$$

Pero, $\operatorname{gr}(g(x)f(x)) < \operatorname{gr}(p(x)f(x)) = n$, y así $g(x)f(x) = 0$, lo cual es imposible. Por lo tanto, $p(x) = g(x)$. □

Para cada q fijo, sea $\mathcal{D}_n$ el conjunto de todos los divisores mónicos de $x^n - 1$, y sea $\mathcal{I}_n$ el conjunto de todos los ideales de R_n, es decir, todos los códigos cíclicos de longitud n. El Teorema 4.3 y la Proposición 4.5 implican que el mapa

$$\Psi : \mathcal{D}_n \to \mathcal{I}_n, \qquad g(x) \mapsto \langle\langle g(x)\rangle\rangle, \tag{4.1}$$

que a cada divisor mónico $g(x)$ de $x^n - 1$ le asocia el código $\langle\langle g(x)\rangle\rangle$ generado por $g(x)$, es una correspondencia biunívoca entre $\mathcal{D}_n$ e $\mathcal{I}_n$.

Observación 4.6. Esto muestra la importancia de poder factorear $x^n - 1$ sobre cuerpos finitos. En efecto, si podemos factorear a x^n-1 completamente sobre $\mathbb{F}_q$, entonces podemos saber cuáles son todos los códigos cíclicos q-arios de longitud n. Podría sin embargo suceder que algunos de éstos sean equivalentes entre sí.

Ejemplo 4.7. Estudiemos los códigos cíclicos binarios de longitud n, para $n \leq 9$ y n impar. Veamos primero que las factorizaciones de $x^n - 1$ en $\mathbb{F}_2[x]$, con $3 \leq n \leq 9$ y n impar, están dadas por

$$\begin{aligned} x^3 - 1 &= (x-1)(x^2 + x + 1), \\ x^5 - 1 &= (x-1)(x^4 + x^3 + x^2 + x + 1), \\ x^7 - 1 &= (x-1)(x^3 + x + 1)(x^3 + x^2 + 1), \\ x^9 - 1 &= (x-1)(x^2 + x + 1)(x^6 + x^3 + 1). \end{aligned} \tag{4.2}$$

Usaremos la siguiente identidad, bien conocida,

$$x^n - 1 = \prod_{d|n} \Phi_d(x),$$

donde $\Phi(x)$ es el d-ésimo *polinomio ciclotómico* de orden n. Por definición, $\Phi(x)$ es el polinomio cuyas raíces son las raíces n-ésimas de la unidad de grado d. Es decir,

$$\Phi_d(x) = \prod_{(k,n)=1} (x - \omega^k),$$

donde ω es una raíz primitiva n-ésima de la unidad de orden d. Se sabe que $\Phi_d(x) \in \mathbb{Z}[x]$ es irreducible sobre $\mathbb{Q}$, y tiene grado $\phi(d)$. Sin embargo, en general, $\Phi_d(x)$ no es irreducible sobre $\mathbb{F}_q$. Por otra parte, notemos que si $n = p$ es primo, entonces

$$x^p - 1 = \Phi_1(x)\Phi_p(x)$$

y, como $\Phi_1(x) = x - 1$, tenemos

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

Veamos (4.2). Es claro que $x^3-1=\Phi_1(x)\Phi_3(x)=(x-1)(x^2+x+1)$ y que ambos polinomios son irreducibles sobre $\mathbb{F}_2$. Del mismo modo, $x^5-1=(x-1)(x^4+x^3+x^2+x+1)$ y $\Phi_5(x)=x^4+x^3+x^2+x+1$ es irreducible sobre $\mathbb{F}_2$. En efecto, como $\Phi_5(x)$ no tiene raíces no tiene factores lineales, y no es producto de dos polinomios cuadráticos, ya que x^2+x+1 es el único irreducible de grado 2 en $\mathbb{F}_2[x]$ y $(x^2+x+1)^2=x^4+x^2+1$.

Ahora, sabemos que $x^7-1=\Phi_1(x)\Phi_7(x)=(x-1)(x^6+x^5+x^4+x^3+x^2+x+1)$. Es fácil chequear que

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

Además, x^3+x^2+1 y x^3+x+1 son irreducibles en $\mathbb{F}_2[x]$, pues no tienen raíces.

Por último, $x^9-1=\Phi_1(x)\Phi_3(x)\Phi_9(x)$. Como conocemos $\Phi_1(x)$ y $\Phi_3(x)$, podemos calcular

$$\Phi_9(x)=\frac{x^9-1}{(x-1)(x^2+x+1)}=x^6+x^3+1.$$

Basta ver que x^6+x^3+1 es irreducible sobre $\mathbb{F}_2$. Como $\Phi_9(x)$ no tiene raíces en $\mathbb{F}_2$, no tiene factores lineales. Luego, si $\Phi_9(x)$ fuera reducible, entonces se factorearía como producto de dos polinomios de grados 2 y 4 respectivamente, o como dos de grado 3. Observar que x^2+x+1 es el único irreducible de grado 2 sobre $\mathbb{F}_2$ y no divide a $\Phi_9(x)$ pues $x^6+x^3+1=(x^2+x+1)(x^4+x^3)+1$. Además, los únicos polinomios irreducibles de grado 3 son x^3+x^2+1 y x^3+x+1, pero vimos que su producto da $\Phi_7(x)$. Por último, $(x^3+x^2+1)^2=x^6+x^4+1$ y $(x^3+x+1)^2=x^6+x^2+1$.

Vimos que si p es primo, entonces $x^p-1=(x-1)(x^{p-1}+\cdots+x+1)$. Si $\Phi_p(x)=x^{p-1}+\cdots+x+1$ resulta ser irreducible sobre $\mathbb{F}_q$, entonces hay 4 códigos cíclicos q-arios de longitud $n=p$, básicamente triviales. Si $q=2$, estos son (ver Proposición 4.13)

$$\begin{aligned}
&C_1(p)=\langle 0\rangle=\{\mathbf{0}\},\\
&C_2(p)=\langle x-1\rangle=\{x\in\mathbb{F}_2 : w(x) \text{ es par}\}=E(p),\\
&C_3(p)=\langle x^{p-1}+\cdots+x+1\rangle=\{00\cdots0, 11\cdots1\}=Rep_2(p),\\
&C_4(p)=\langle x^p-1\rangle=\mathbb{F}_2^p.
\end{aligned}\tag{4.3}$$

Por las expresiones en (4.2), sabemos que hay $2^2=4$ códigos cíclicos sobre $\mathbb{F}_2$ de longitudes 3 y 5, respectivamente, y que hay $2^3=8$ códigos cíclicos de longitudes 7 y 9, respectivamente. En R_3 y R_5 son los ya mencionados en (4.3), con $p=3$ y $p=5$, respectivamente.

En R_7, además de los mencionados en (4.3), hay 4 códigos más, ya que $\Phi_7(x)$ se parte como producto de dos polinomios. Estos son,

$C_1(7)=\langle 0\rangle=\{\mathbf{0}\}$,
$C_2(7)=\langle x-1\rangle=E(7)$,
$C_3(7)=\langle x^3+x+1\rangle$,
$C_4(7)=\langle x^3+x^2+1\rangle$,
$C_5(7)=\langle (x-1)(x^3+x+1)\rangle=\langle x^4+x^3+x+1\rangle$,
$C_6(7)=\langle (x-1)(x^3+x^2+1)\rangle=\langle x^4+x^2+x+1\rangle$,
$C_7(7)=\langle (x^3+x+1)(x^3+x^2+1)\rangle=\langle x^6+x^5+x^4+x^3+x^2+x+1\rangle=\{\mathbf{0},\mathbf{1}\}=Rep_2(7)$,
$C_8(7)=\langle x^7-1\rangle=\mathbb{F}_2^7$.

Veamos, por ejemplo, el código $C_3(7)=\langle x^3+x+1\rangle$. Para obtener este código más fácilmente, armamos la siguiente tabla, en donde en la primer columna ponemos los factores $p(x)$ y en la segunda los múltiplos $p(x)(x^3+x+1)$ del polinomio generador. Como multiplicar por x es hacer un desplazamiento cíclico en las palabras de $\mathbb{F}_2^7$, las palabras códigos de $C_3(7)$ se obtienen haciendo los desplazamientos del polinomio generador por 1, x, x^2 y x^3, y luego haciendo todas las posibles sumas de estas cuatro palabras.

$p(x)$	$p(x)(x^3+x+1)$	palabra código
0	0	0000000
1	x^3+x+1	1101000
x	x^4+x^2+x	0110100
x^2	$x^5+x^3+x^2$	0011010
x^3	$x^6+x^4+x^3$	0001101
$1+x$	$x^4+x^3+x^2+1$	1011100
$1+x^2$	x^5+x^2+x+1	1110010
$1+x^3$	x^6+x^4+x+1	1100101
$x+x^2$	$x^5+x^4+x^3+x$	0101110
$x+x^3$	$x^6+x^3+x^2+x$	0111001
x^2+x^3	$x^6+x^5+x^4+x^2$	0010111
$1+x+x^2$	x^5+x^4+1	1000110
$1+x+x^3$	x^6+x^2+1	1010001
$1+x^2+x^3$	$x^6+x^5+x^4+x^3+x^2+x+1$	1111111
$x+x^2+x^3$	$x^6+x^5+x^2$	0100011
$1+x+x^2+x^3$	$x^6+x^5+x^3+1$	1001011

Notar que el código $C_3(7)$ coincide con el código de Hamming binario $\mathcal{H}_2(3)$ del Ejemplo 1.10. Es decir, $\mathcal{H}_2(3)$ es cíclico.

Similarmente, obtenemos el código $C_4(7) = \langle x^3+x^2+1 \rangle$

$p(x)$	$p(x)(x^3+x^2+1)$	palabra código
0	0	0000000
1	x^3+x^2+1	1011000
x	x^4+x^3+x	0101100
x^2	$x^5+x^4+x^2$	0010110
x^3	$x^6+x^5+x^3$	0001011
$1+x$	x^5+x^2+x+1	1110100
$1+x^2$	$x^5+x^4+x^3+1$	1001110
$1+x^3$	$x^6+x^5+x^2+1$	1010011
$x+x^2$	$x^5+x^3+x^2+x$	0111010
$x+x^3$	$x^6+x^5+x^4+x$	0100111
x^2+x^3	$x^6+x^4+x^3+x^2$	0011101
$1+x+x^2$	x^5+x+1	1100010
$1+x+x^3$	$x^6+x^5+x^4+x^3+x^2+x+1$	1111111
$1+x^2+x^3$	x^6+x^4+1	1000101
$x+x^2+x^3$	x^6+x^2+x	0110001
$1+x+x^2+x^3$	x^6+x^3+x+1	1101001

Este código es equivalente a $C_3(7) = \mathcal{H}_2(3)$ (ejercicio).

Los códigos $C_5(7)$ y $C_6(7)$ restantes pueden ser obtenidos de la misma forma que antes, o también, a partir de sus matrices generadoras

$$G_5(7) = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}, \qquad G_6(7) = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Dejamos los detalles de esto a cargo del lector.

¿Se anima a calcular todos los códigos cíclicos binarios de longitud 9 (ejercicio)? □

Del ejemplo anterior surgen los siguientes interrogantes.

Pregunta 4.4. ¿Son los códigos de Hamming binarios $\mathcal{H}_2(r)$ cíclicos o equivalentes a códigos cíclicos? ¿Son los códigos de Hamming q-arios $\mathcal{H}_q(r)$ cíclicos o equivalentes a códigos cíclicos?

Se puede probar que todos los códigos de Hamming binarios $\mathcal{H}_2(r)$ son equivalentes a códigos cíclicos. En el caso ganeral, sin embargo, el código $\mathcal{H}_q(r)$ es equivalente a un código cíclico sólo si $(r, q-1) = 1$.

Ejemplo 4.8. Por el comentario anterior, los códigos de Hamming ternarios $\mathcal{H}_3(2m+1)$ son equivalentes a códigos cíclicos. Para $\mathcal{H}_3(2m)$ tenemos $(r, q-1) = 2$, y por lo tanto no podemos asegurar nada.

Por ejemplo, $\mathcal{H}_3(2)$ no es equivalente a un código cíclico. En efecto, $\mathcal{H}_3(2)$ tiene parámetros

$$n = \tfrac{3^2-1}{3-1} = 4, \qquad k = n - r = 2, \qquad d = 3.$$

Supongamos que C es un código cíclico ternario de longitud 4, es decir C es un ideal de $\mathbb{F}_3[x]/\langle x^4 - 1\rangle$. Sobre $\mathbb{F}_3$ tenemos la factorización en irreducibles

$$x^4 - 1 = (x-1)(x+1)(x^2+1).$$

De los $2^3 = 8$ códigos cíclicos ternarios de longitud 4, hay solamente dos con $k = 2$, estos son

$$C_1 = \langle x^2+1\rangle \quad \text{y} \quad C_2 = \langle (x-1)(x+1)\rangle = \langle x^2-1\rangle = \langle x^2+2\rangle.$$

Como $w(x^2+1) = w(1010) = 2$ y $w(x^2+2) = w(2010) = 2$, en ambos casos tenemos que $w_C \leq 2$ y por lo tanto $d \neq 3$. Luego $\mathcal{H}_3(2)$ no es equivalente a un código cíclico. ¿Es $\mathcal{H}_3(4)$ equivalente a un código cíclico (ejercicio)? □

Los códigos de Golay también son equivalentes a códigos cíclicos.

Ejemplo 4.9. Se puede ver que $x^{23} - 1$ se factorea en irreducibles sobre $\mathbb{F}_2$ de la siguiente manera

$$x^{23} - 1 = (x+1)(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1).$$

El código $C_1 = \langle x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1\rangle$ tiene parámetros $[23, 12]$. Se puede mostrar (no es difícil, ver el libro de Hill ([3])) que C_1 tiene distancia mínima $d = 7$. Luego, C_1 tiene los mismos parámetros que el código de Golay $\mathcal{G}_{23}$. Así, por la unicidad de los parámetros de los códigos de Golay, C_1 es equivalente a $\mathcal{G}_{23}$, es decir, $\mathcal{G}_{23}$ es equivalente a un código cíclico.

Similarmente, sobre $\mathbb{F}_3$ tenemos

$$x^{11} - 1 = (x-1)(x^5 + x^4 - x^3 + x^2 - 1)(x^5 - x^3 + x^2 - x - 1).$$

El código $C_2 = \langle x^5 + x^4 - x^3 + x^2 - 1\rangle$ tiene parámetros $[11, 6, 5]$ y por lo tanto es equivalente al código ternario de Golay $\mathcal{G}_{11}$. □

Los ejemplos anteriores muestran que los códigos perfectos que conocemos son cíclicos.

Notar que si C_1 y C_2 son códigos cíclicos en R_n, entonces la *suma*

$$C_1 + C_2 = \{c_1 + c_2 : c_1 \in C_1, c_2 \in C_2\}$$

es el menor código cíclico que contiene a C_1 y a C_2.

La prueba del siguiente resultado se deja como ejercicio para el lector.

Proposición 4.10. *Sean $C_1 = \langle\langle g_1(x)\rangle\rangle$ y $C_2 = \langle\langle g_2(x)\rangle\rangle$ códigos cíclicos en R_n. Entonces,*

(1) *$C_1 \subset C_2$ si y sólo si $g_2(x) \mid g_1(x)$,*
(2) *$C_1 \cap C_2 = \langle\langle m.c.m\{g_1(x), g_2(x)\}\rangle\rangle$,*
(3) *$C_1 + C_2 = \langle\langle m.c.d\{g_1(x), g_2(x)\}\rangle\rangle$.*

Este resultado dice que el mapa $\Psi : g(x) \to \langle\langle g(x)\rangle\rangle$ en (4.1) es un isomorfismo que invierte orden entre los retículos (lattices) $(\mathcal{D}_n, |)$ y $(\mathcal{I}_n, \subset)$.

4.2. Polinomio de chequeo. Como el polinomio generador $g(x)$ de un $[n, n-r]$-código cíclico en R_n divide a $x^n - 1$, tenemos

$$x^n - 1 = g(x)h(x),$$

donde $h(x)$ es un polinomio de grado $n-r$, llamado *polinomio de chequeo* o *de control* (check polynomial) de C. Tenemos el siguiente resultado que resume las propiedades de $h(x)$.

Teorema 4.11. *Sea $h(x)$ el polinomio de chequeo de un código cíclico C en R_n.*

(1) *El código C puede describirse como*

$$C = \{p(x) \in R_n : p(x)h(x) \equiv 0\}.$$

(2) *Si $h(x) = h_0 + h_1x + \cdots + h_{n-r}x^{n-r}$, entonces la matriz de control de paridad de C está dada por*

$$H = \begin{pmatrix} h_{n-r} & \cdots & \cdots & h_0 & 0 & 0 & \cdots & 0 \\ 0 & h_{n-r} & \cdots & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & 0 & h_{n-r} & \cdots & \cdots & h_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & h_{n-r} & \cdots & \cdots & h_0 \end{pmatrix}$$

(3) *El código dual $C^\perp$ es el código cíclico de dimensión r con polinomio generador*

$$h^\perp(x) = h_0^{-1}x^{n-r}h(x^{-1}) = h_0^{-1}(h_0x^{n-r} + h_1x^{n-r+1} + \cdots + h_{n-r}).$$

Demostración. (1) Sea $g(x)$ el polinomio generador de C. Si $p(x) \in C$, entonces $p(x) = f(x)g(x)$ para algún $f(x) \in R_n$. Luego,

$$p(x)h(x) = f(x)g(x)h(x) = f(x)(x^n - 1) \equiv 0.$$

Por otra parte, si $p(x) \in R_n$ y $p(x)h(x) \equiv 0$, tenemos $p(x) = q(x)g(x)+r(x)$, con $\operatorname{gr}(r(x)) < r$. Entonces,

$$p(x)h(x) = q(x)g(x)h(x) + r(x)h(x),$$

de donde $r(x)h(x) \equiv 0$. Sin embargo, $\operatorname{gr}(r(x)h(x)) < r - (n-r) = n$, por lo que $r(x)h(x) = 0$. Luego, $r(x) = 0$ y $p(x) = q(x)g(x) \in C$.

(2) Si $c(x) \in C$, entonces $c(x)h(x) \equiv 0$. Ahora, como $\operatorname{gr}(c(x)h(x)) < 2n - r$, deducimos que los coeficientes de $x^{n-r}, x^{n-r+1}, \ldots, x^{n-1}$, en el producto $c(x)h(x)$ son 0, es decir,

$$\begin{aligned} c_0h_{n-r} + c_1h_{n-r+1} + \cdots + c_{n-r}h_0 &= 0 \\ c_1h_{n-r} + c_2h_{n-r+1} + \cdots + c_{n-r+1}h_0 &= 0 \\ &\vdots \\ c_{r-1}h_{n-r} + c_rh_{n-r+1} + \cdots + c_{n-1}h_0 &= 0. \end{aligned}$$

Pero esto es equivalente a $(c_0c_1 \ldots c_{n-1})H^\intercal = 0$, y así H genera un código C' que es ortogonal a C, o sea, $C' \subset C^\perp$. Como $h_{n-r} \neq 0$, sigue que $\dim C = r$, y por lo tanto $C' = C^\perp$.

(3) Si vemos que $h^\perp(x)$ divide a $x^n - 1$, entonces sabemos que es el polinomio generador de un código cíclico $\langle h^\perp(x)\rangle$ con matriz generadora H, y así $\langle h^\perp(x)\rangle = C^\perp$. Pero $h(x)g(x) = x^n - 1$ implica $h(x^{-1})g(x^{-1}) = x^{-n} - 1$, o sea

$$x^{n-r}h(x^{-1})g(x^{-1}) = 1 - x^n,$$

de donde sale que $h^\perp(x) \mid x^n - 1$. □

Ejemplo 4.12. El código $C = \langle x^3 + x + 1\rangle = \mathcal{H}_2(3)$, visto en el Ejemplo 4.7, tiene polinomio de chequeo

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

y como

$$h^\perp(x) = x^4 h(x^{-1}) = x^4(x^{-4} + x^{-2} + x^{-1} + 1) = 1 + x^2 + x^3 + x^4,$$

el código C tiene matriz de paridad

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

Proposición 4.13. *Sea $E(n)$ el código que consta de todas las palabras de peso par en $\mathbb{F}_2^n$, y sea C un código cíclico binario de longitud n. Entonces,*

(1) $E(n) = \langle\langle x - 1\rangle\rangle$.

(2) $C = \langle\langle g(x)\rangle\rangle \subset E(n)$ *si y sólo si* $x - 1 \mid g(x)$.

Demostración. Para ver la parte (1), observamos que para el código cíclico $\langle\langle x-1\rangle\rangle$ tenemos

$$h(x) = \frac{x^n - 1}{x - 1} = x^{n-1} + x^{n-2} + \cdots + 1 = h^\perp(x)$$

y por lo tanto

$$\langle\langle x-1\rangle\rangle = \langle\langle x-1\rangle\rangle^{\perp\perp} = \{\mathbf{0}, \mathbf{1}\} = E(n).$$

La parte (2) sigue de la parte (1) y de la Proposición 4.10. □

4.3. Codificación y decodificación de códigos cíclicos. Existen dos maneras bastante directas de codificar mensajes usando códigos cíclicos, una sistemática y otra no-sistemática. Sea $C = \langle\langle g(x)\rangle\rangle$ un $[n, n-r]$-código cíclico q-ario, con $\mathrm{gr}(g(x)) = r$. Entonces, C puede codificar mensajes q-arios de longitud $n - r$ y requiere r símbolos de redundancia.

Codificación No-sistemática. Dado un mensaje $a_0 a_1 \ldots a_{n-r-1}$ en $\mathbb{F}_q^{n-r}$, formamos el *polinomio mensaje*

$$a(x) = a_0 + a_1 x + \cdots + a_{n-r-1} x^{n-r+1}.$$

Este polinomio se codifica como el producto $c(x) = a(x)g(x) \in C$, es decir

$$a(x) \rightsquigarrow a(x)g(x).$$

Codificación Sistemática. Para obtener una codificación sistemática, formamos el polinomio mensaje

$$\overline{a}(x) = a_0 x^{n-1} + a_1 x^{n-2} + \cdots + a_{n-r-1} x^r.$$

Notar que $\overline{a}(x)$ no tiene términos de grado menores que r. Ahora, dividimos $\overline{a}(x)$ por $g(x)$,

$$\overline{a}(x) = q(x)g(x) + r(x) \qquad \text{con } \mathrm{gr}(r(x)) < r,$$

y mandamos el código $c(x) = \overline{a}(x) - r(x) = q(x)g(x) \in C$. Es decir, codificamos

$$\overline{a}(x) \rightsquigarrow q(x)g(x).$$

Como $a(x)$ y $r(x)$ tienen términos de distintos grados, la codificación es sistemática. En efecto, si leemos los términos de un polinomio desde el de grado máximo hasta el de grado mínimo, vemos que las primeras $n - r$ coordenadas son símbolos de información, mientras que las restantes son símbolos de redundancia.

Ejemplo 4.14. Consideremos el $[7,4,3]$-código de Hamming $C = \mathcal{H}_2(3)$ como código cíclico, es decir generado por $g(x) = x^3 + x + 1$ (ver Ejemplo 4.7). Consideremos el mensaje 1010. Usando la codificación no-sistemática formamos $a(x) = 1 + x^2$ y codificamos como

$$c(x) = a(x)g(x) = (x^2 + 1)(x^3 + x + 1) = x^5 + x^2 + x + 1,$$

es decir

$$1010 \rightsquigarrow 1110010.$$

Ahora usemos codificación sistemática. Formamos el polinomio $\overline{a}(x) = x^6 + x^4$ y dividimos $\overline{a}(x) = (x^3 + 1)(x^3 + x + 1) + (x + 1)$. Luego, codificamos $c(x) = \overline{a}(x) - r(x) = x^6 + x^4 + x + 1$, es decir

$$1010 \rightsquigarrow 1100\underleftarrow{101}.$$

Leyendo esta palabra código 1100101 de atrás para adelante, las primeras 4 coordenadas dan la palabra mensaje 1010. □

Decodificación. Como todo código cíclico es lineal, podemos usar decodificación por síndrome, pero en su forma polinómica. Si $c(x) \in C$ es el código enviado y $u(x)$ es el polinomio recibido entonces $e(x) = u(x) - c(x)$ es el *polinomio error*. El peso de un polinomio es el número de coeficientes no-nulos.

Definición 4.15. Sea $C = \langle\langle g(x) \rangle\rangle$ un $[n, n-r]$-código cíclico. El *síndrome* de un polinomio $u(x)$, denotado por $syn(u(x))$, es el resto de la división $u(x)$ por $g(x)$, es decir

$$u(x) = q(x)g(x) + syn(u(x)), \qquad \mathrm{gr}(syn(u(x))) < r.$$

Se puede ver que esta definición de síndrome coincide con la definición de síndrome dada para códigos lineales.

Un polinomio recibido $u(x)$ es una palabra código si y sólo si su síndrome es el polinomio nulo. Además, dos polinomios tienen el mismo síndrome si y sólo si están en la misma coclase de C. Luego, la forma polinómica de decodificación por síndrome es análoga a la forma vectorial.

Ejemplo 4.16. Como el código de Hamming $\mathcal{H}_2(3)$ es 1-corrector, es capaz de corregir todos los polinomios error de peso a lo sumo 1. Luego, los líderes de coclase y sus síndromes son

Líderes	Síndromes
0	0
1	1
x	x
x^2	x^2
x^3	$x + 1$
x^4	$x^2 + x$
x^5	$x^2 + x + 1$
x^6	$x^2 + 1$

Si, por ejemplo, recibimos el polinomio $u(x) = x^6 + x + 1$, calculamos su síndrome

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

Como $syn(u(x)) = x^2 + x$, por la tabla de arriba vemos que el líder de coclase es $a(x) = x^4$ y, por lo tanto, decodificamos $u(x)$ como

$$c(x) = u(x) - a(x) = x^6 + x^4 + x + 1 = 1100101 \in C.$$

□

Hasta ahora hemos usado que C es lineal, pero no que es cíclico. Veamos como podemos sacar provecho de este hecho para mejorar el proceso de decodificación. Supongamos que tenemos algún método para decodificar el coeficiente principal de cualquier palabra recibida $u(x)$, que a los fines de la siguiente discusión podemos suponer que es el coeficiente de x^{n-1}, sin preocuparnos si este es no-nulo o no. Luego, podemos decodificar el coeficiente principal de $u(x)$, realizar un desplazamiento cíclico módulo $x^n - 1$, y decodificar el nuevo coeficiente principal, que es el coeficiente de x^{n-2} en $u(x)$. Repitiendo este proceso decodificamos toda la palabra. Este método ahorra tiempo porque sólo necesitamos las filas de la tabla de líderes-síndromes que contienen líderes de grado $n-1$.

Ejemplo 4.17. Volviendo al ejemplo anterior, como el único líder de peso 1 y grado $n-1=6$ es x^6, sólo necesitamos la tabla

Líder	Síndrome
x^6	x^2+1

Supongamos que, como antes, recibimos $u(x) = x^6+x+1$. Como $\mathrm{syn}(u(x)) = x^2+x$ no está en la tabla, asumimos que el coeficiente de $u(x)$ es correcto. Le realizamos un desplazamiento a $u(x)$,

$$x(x^6+x+1) \mod (x^7-1) = x^2+x+1$$

y calculamos su síndrome, que es justamente x^2+x+1. Como este no está en la tabla asumimos que el coeficiente de x^5 es correcto. Desplazando una vez más y calculando síndromes tenemos que

$$x(x^2+x+1) = x^3+x^2+x = 1\cdot(x^3+x+1)+(x^2+1).$$

Como el síndrome x^2+1 está en la tabla, deducimos que el coeficiente de x^4 en $u(x)$ es incorrecto. Continuando de esta manera, decodificamos $u(x)$ como $c(x) = x^6+x^4+x+1$. □

4.4. Ceros de polinomios y códigos cíclicos famosos. Existe una forma alternativa de ver a los códigos cíclicos en R_n. Éstos, pueden ser caracterizados por ceros del polinomio x^n-1, es decir, por ciertas raíces n-ésimas de la unidad. Sea

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

la factorización de x^n-1 en factores irreducibles mónicos sobre $\mathbb{F}_q$. Si α es una raíz de $m_i(x)$ en algún cuerpo extensión de $\mathbb{F}_q$, entonces $m_i(x)$ es el polinomio minimal de α sobre $\mathbb{F}_q$. Luego, si $f(x) \in \mathbb{F}_q[x]$, entonces $f(\alpha) = 0$ si y sólo si $f(x) = a(x)m_i(x)$ para algún $a(x)$. En particular, si $f(x) \in R_n$, entonces

$$f(\alpha) = 0 \text{ si y sólo si } f(x) \in \langle\langle m_i(x) \rangle\rangle.$$

Generalizando, tenemos el siguiente resultado.

Teorema 4.18. *Sea $g(x) = q_1(x)q_2(x)\cdots q_t(x)$ un producto de factores irreducibles de x^n-1, y sean $\{\alpha_1,\ldots,\alpha_s\}$ las raíces de $g(x)$ en el cuerpo de descomposición de x^n-1 sobre $\mathbb{F}_q$. Entonces*

$$\langle\langle g(x)\rangle\rangle = \{f(x) \in R_n : f(\alpha_1) = 0, \ldots, f(\alpha_s) = 0\}.$$

Más aún, es suficiente tomar una raíz de cada factor irreducible de $g(x)$. Esto es, si β_i es una raíz de $q_i(x)$ para $i = 1,\ldots,t$, entonces

$$\langle\langle g(x)\rangle\rangle = \{f(x) \in R_n : f(\beta_1) = 0, \ldots, f(\beta_t) = 0\}.$$

Las raíces del polinomio generador de un código cíclico se denominan *ceros* del código. La descripción de códigos cíclicos a través de sus ceros permite definir muchas familias famosas de códigos cíclicos, por ejemplo: *residuos cuadráticos*, *BCH*, *Reed-Solomon*, *alternantes* y de *Goppa*. Pero esto mejor lo dejamos para otro curso...

Apéndice A: Status Tecnológico de Códigos

Listamos a continuación algunos de los códigos más utilizados en aplicaciones tecnológicas por empresas como IBM, Philips y Sony, y por agencias espaciales como la NASA y la Agencia Espacial Europea.

- Códigos de Reed-Muller $\mathcal{R}(1,5)$.
- Códigos de Golay extendido $\mathcal{G}_{24}$.
- Códigos de Reed-Solomon.
- Códigos de convolución $(2,1)M = 6$ y $(2,1)M \geq 24$.
- Códigos de Fire acortado. [IBM, discos duros].
- Códigos de residuos cuadráticos $\mathcal{QR}(48,24)$.
- Concatenación de códigos de convolución y de Reed-Solomon.
- Intercalación cruzada de códigos de Reed-Solomon $\mathcal{RS}(28,24)$ y $\mathcal{RS}(32,28)$ (Cross-interleaved Reed-Solomon Codes o CIRC). [Philips y Sony, discos compactos].

Para finalizar, presentamos una breve reseña histórica de algunas de las misiones espaciales y los códigos que en ellas se utilizaron para transmitir fotografías.

⋆ [**1965**, NASA, Mariner 4.]
La sonda Mariner 4 fue la primera en tomar fotografías de otro planeta. En 1965 tomo 22 fotografías completas de Marte en blanco y negro, cada una de 200×200 pixeles. A cada pixel se le asignó una 6-upla representando uno de 64 tonos de gris según el brillo, desde el blanco 000000 hasta el negro 111111. Se utilizó un código binario de Hadamard con parámetros $[32,6,7]$ y la transformada rápida de Fourier (FFT) para la decodificación. El número total de bits (dígitos binarios) por fotografía es de 240.000. La transmisión se llevó a cabo a un ritmo de $8\frac{1}{3}$ bits por segundo, por lo que llevó ¡exactamente 8 horas transmitir cada foto! (controlar).

⋆ [**1969-1972**, NASA, Mariner 6–9, Reed-Muller.]
Desde 1969 a 1972, las sondas Mariner 6, 7 y 9 enviaron fotografías de Marte. En enero de 1972, la sonda Mariner 9 tomó fotografías en blanco y negro del planeta Marte esta vez de $600 \times 600 = 360.000$ pixeles. Se utilizó un código de Reed-Muller $\mathcal{R}(1,5)$ con parámetros $(32, 2^6, 16)$. Este código corrige 7 errores y simultáneamente detecta 8 errores, con $R = \frac{3}{16}$. A cada pixel se le asignó una 6-upla representando el brillo. Ahora, cada pixel fue codificado como una palabra de longitud 32 (26 bits de redundancia). La tasa de transmisión fue aumentada de $8\frac{1}{3}$ a 16.200 bits por segundo. Sin embargo, las cámaras tomaban imágenes a razón de más de 100.000 bits por segundo, por lo que los datos debieron ser almacenados en cintas magnéticas antes de ser transmitidos.

⋆ [**1976**, NASA, códigos de residuos cuadráticos.]
En 1976 el *Jet Propulsion Laboratory* de la NASA diseñó un algoritmo de decodificación para un código de residuos cuadráticos con parámetros $[48,24]$.

⋆ [**1976**, NASA, Viking 1]
La sonda Viking 1 amariza y envía fotograías a color de Marte de alta calidad.

⋆ [**1977**, NASA, códigos de Reed-Solomon.]
Desde 1977, la NASA utilizó los códigos de Reed-Solomon en las misiones Galileo, Magallanes y Ulises.

⋆ [**1977**, NASA, códigos de convolución.]
Desde 1977, la NASA comenzó a usar códigos convolucionales. Misiones espaciales como las *Pioneer* de la NASA y la alemana *Helios* usaron decodificación secuencial en códigos de convolución grandes.

⋆ [**1979-1981**, NASA, Voyager 1, códigos de Golay.]
Desde 1979 hasta 1981 la sonda Voyager 1 tomó fotos color de alta resolución de Júpiter y Saturno. Se usaron 4096 tonos distintos de color y se codificaron los mensajes con el código de Golay extendido $\mathcal{G}_{24}$ con parámetros $(24, 4096, 8)$.

⋆ [**1986-1989**, NASA, Voyager 2, códigos de Reed-Solomon.]
Entre 1986 y 1989, la sonda Voyager 2 tomó fotos en color de alta calidad de Urano y Neptuno usando códigos de Reed-Solomon. Esta sonda transmitía 115.200 bits por segundo!

Apéndice B: El código ISBN

Veamos el código ISBN (*International Standard Book Number*) utilizado para nomenclar los libros publicados en todo el mundo. Cada libro es individualizado por una secuencia de 10 dígitos. Los primeros 9 dígitos representan el idioma en que el libro está escrito, la editorial que lo publicó y un número que la editorial le asigna al libro. El último dígito es de control. Por ejemplo, mirando en mi biblioteca, encuentro que

- "*Coding and Information Theory*", de Steve Roman, tiene ISBN 0-387-97812-7, y que
- "*Crónicas del Angel Gris*", de Alejandro Dolina, tiene ISBN 950-581-693-6.

En los ejemplos anteriores, el 0 representa al inglés y 387 a la editorial Springer-Verlag, mientras que 950 representa al español (¿de Argentina?) y 581 a Ediciones Colihue.

El último dígito x_{10} se toma de modo que todo el número $x_1x_2\dots x_{10}$ satisfaga

$$\sum_{i=1}^{10} ix_i \equiv 0 \mod (11).$$

Es decir, es un *control de paridad con peso*. Como $10x_{10} \equiv -x_{10} \mod (11)$, el código ISBN queda definido por

$$\mathcal{I} = \{x_1x_2\dots x_{10} : 0 \le x_1, \dots, x_9 \le 9 \quad \text{y} \quad x_{10} = \sum_{i=1}^{9} ix_i \mod (11)\}.$$

En caso que $x_{10} = 10$ se escribe X en lugar de 10. Por ejemplo, "*Ficciones*", de Jorge Luis Borges, tiene

ISBN 950-04-0205-X.

En efecto, tenemos

$$9 + 2\cdot 5 + 5\cdot 4 + 7\cdot 2 + 9\cdot 5 = 9 + 10 + 20 + 14 + 45 = 98 = 88 + 10 \equiv 10 \mod (11),$$

y esto explica porqué va la X.

El código ISBN no es lineal, es un subconjunto de $\mathbb{F}_{11}^{10}$ ¿Qué podemos decir sobre las propiedades autocorrectoras de este código? Este código permite:

(1) Detectar 1 error, aunque no sepamos donde está (luego hay que pedir retransmisión del mensaje). En efecto, supongamos que al enviar $x = x_1\dots x_{10}$ se recibe y con un único error en la coordenada j, es decir, $y_i = x_i$ para $i \neq j$ y $y_j = x_j + a$ con $a \neq 0$. Luego,

$$\sum_{i=1}^{10} iy_i = (\sum_{i=1}^{10} ix_i) + ja = ja \not\equiv 0 \mod (11),$$

pues $\mathbb{F}_{11}$ es un cuerpo.

(2) Detectar 2 errores obtenidos por transposición de símbolos. Si ahora recibimos y, que es igual a x salvo que los símbolos x_j y x_k están permutados entre sí, entonces

$$\sum_{i=1}^{10} iy_i = \sum_{i=1}^{10} ix_i + (k-j)x_j + (j-k)x_k = (k-j)(x_j - x_k) \not\equiv 0 \mod (11),$$

si $k \neq j$ y $x_j \neq x_k$.

(3) Corregir 1 error, sabiendo dónde está el error. Si recibimos $x = x_1 \dots x_{10}$ y sabemos que la coordenada x_j es errónea, entonces podemos despejar x_j de

$$x_1 + \cdots + jx_j + \cdots + 10x_{10} = 0$$

y tenemos

$$x_j = j^{-1} \sum_{i \neq j} ix_i,$$

donde nuevamente usamos que $\mathbb{F}_{11}$ es cuerpo!

Por ejemplo, supongamos que un colega nos solicita por fax hacer el pedido de compra de un libro, digamos el "Algebra" de Lang. Este nos envía los datos, pero el fax llega borroso y recibimos ISBN 0-201-555z0-9. Nos damos cuenta que hubo un error, pero no desesperamos, y hacemos

$$0 = 2 \cdot 2 + 4 \cdot 1 + 5 \cdot 5 + 6 \cdot 5 + 7 \cdot 5 + 8z + 10 \cdot 9 = 4 + 4 + 25 + 30 + 35 + 8z + 90$$

de donde $0 = 8z + 188 = 8z + 8 \cdot 17 + 1 \equiv 8z + 1 \mod (11)$. Luego, como $8 \cdot 7 = 56 \equiv 1 \mod (11)$, tenemos

$$z = (-1) \cdot 8^{-1} = -7 \equiv 4 \mod (11).$$

Entonces, hacemos el pedido de la compra del libro, con el código correcto ISBN 0-201-55540-9.

Ejercicios

Sobre códigos en general.

1. Explicar porqué el código C_3 de la Introducción detecta hasta dos errores y corrige sólo uno.
2. Probar la expresión (1.4), es decir $d(x,y) = w(x) + w(y) - 2w(x \cap y)$.
3. Probar que el código binario $E(n) = \{x \in \mathbb{F}_2^n : w(x) \equiv 0 \,(\mathrm{mod}\, 2)\}$ es lineal con parámetros $[n, n-1, 2]$.
4. Mostrar que el código ternario $C = \{012, 120, 201\}$ no es múltiplo escalar equivalente al código $Rep_3(3) = \{000, 111, 222\}$.
5. Completar los detalles de la demostración del Teorema 1.23. Es decir, probar que si C es un código sobre el alfabeto $\mathcal{A}$ con distancia par $d = 2t + 2$, entonces las esferas de radio $r = t + 1$ cubren $\mathcal{A}^n$ pero no son disjuntas.
6. Chequear que las siguientes familias de parámetros (n, M, d) satisfacen la condición de empaquetamiento de esferas (1.11).
   a) $(n, q^n, 1)$,
   b) $(n, 1, 2n+1)$,
   c) $(2m+1, 2, 2m+1)$,
   d) $(\frac{q^r-1}{q-1}, q^{n-r}, 3)$ con $(r \geq 2)$,
   e) $(23, 2^{11}, 7)$,
   f) $(11, 3^6, 5)$.
7. Completar los detalles de la demostración del Teorema 1.28.
8. Demostrar el Teorema 1.30.

9. Sea C un (n, M, d)-código. Convencerse de que el código extendido
$$\hat{C} = \{c_1c_2 \dots c_nc_{n+1} : \ c_1c_2 \dots c_n \in C \ \text{ y } \ \sum_{k=1}^{n+1} c_k = 0\}$$
es un $(n+1, M, \hat{d})$-código con $\hat{d} = d$ ó $d+1$. Probar que si C es lineal entonces $\hat{C}$ también lo es, y que si C es binario, entonces $\hat{C} \subset E(n+1)$.
10. Probar que si L es un código lineal, el código pinchado L^* también es lineal.
11. Probar que $\mathbb{F}_{q\,\{x_1=0\}}^n = \mathbb{F}_q^{n-1}$ y que $E(n)_{\{x_1=0\}} = E(n-1)$.

Sobre códigos lineales.

1. ¿Puede un $(11, 24, 5)$-código ser lineal?
2. Si C_1 y C_2 tienen matrices generadoras G_1 y G_2 entonces el código pegado tiene matriz generadora $G = (G_1|G_2)$.
3. Sea L un (n, M, d)-código lineal binario. Probar que si L contiene al menos una palabra de peso impar entonces la mitad de las palabras de L son de peso impar.
4. Sea L un código lineal binario. Mostrar que si $\mathbf{1} = 11 \dots 1 \in L$ entonces $L = L^c$, y que si $\mathbf{1} \notin L$ entonces $L \cap L^c = \emptyset$.
5. Demostrar la Proposición 1.38.
6. Encontrar la distancia mínima del código lineal binario con matriz generadora
$$G = \begin{pmatrix} 1&0&0&1&1 \\ 0&0&1&0&1 \\ 0&1&1&1&1 \end{pmatrix}.$$
7. Encontrar la distancia mínima del código lineal ternario con matriz generadora
$$G = \begin{pmatrix} 0&1&2&1 \\ 1&0&2&2 \end{pmatrix}.$$
8. Construir un $[6, 3, 4]$-código sobre $\mathbb{F}_5$.
9. Si un código lineal L tiene matriz de paridad H, ¿cuál es la matriz de paridad $\hat{H}$ del código extendido $\hat{L}$?
10. Probar que si $L = E(n)$ entonces $L^\perp = Rep_2(n)$.
11. Sea $\eta : \mathcal{H}_2(r) \to \mathbb{Z}_2$ un mapa no lineal con $\eta(0) = 0$ y para cada $x \in \mathbb{Z}_2^n$ sea $\varepsilon(x) = 0$, si x tiene peso par, y $\varepsilon(x) = 1$, si x tiene peso impar. Consideremos el código
$$\mathcal{V} = \{(x, x+c, \varepsilon(x) + \eta(c)) : x \in \mathbb{Z}_2^n, c \in \mathcal{H}_2(r)\},$$
donde $n = 2^r - 1$. Probar que $\mathcal{V}$ es un código binario no-lineal con parámetros de Hamming $(2^{r+1} - 1, 2^{2n-r}, 3)$ y, por lo tanto, perfecto.
12. Escribir matrices de paridad para $\mathcal{H}_2(4)$, $\mathcal{H}_3(3)$, $\mathcal{H}_3(4)$, $\mathcal{H}_5(2)$ y $\mathcal{H}_5(3)$.
13. Construir un tabla de síndromes para el código de Hamming $\mathcal{H}_2(3)$ y decodificar las palabras 0000010, 1111111, 1100110 y 1010101.
14. Probar que $\mathbf{1} \in \mathcal{G}_{24}$.
15. Encontrar el enumerador de peso $W(s)$ del código cuya matriz generadora es
$$G = \begin{pmatrix} 1&0&0&1&1 \\ 0&1&0&0&1 \\ 0&0&1&0&1 \end{pmatrix},$$
(a) directamente y (b) usando la identidad de MacWilliams.
16. Sea L un código lineal binario y L_0 el subcódigo que consta de las palabras de peso par en L, es decir $L_0 = \{c \in L : w(c) \equiv 0 \mod (2)\}$. Mostrar que
$$W_{L_0}(s) = \tfrac{1}{2}\big(W_L(s) + W_L(-s)\big).$$
17. Consideremos el código de Golay $\mathcal{G}_{24}$. Decodifique, si es posible, las siguientes palabras recibidas

a) $x = 101111\,101111\,010010\,010010$.
b) $x = 001001\,001101\,101000\,101000$.
c) $x = 000111\,000111\,011011\,010000$.
d) $x = 111000\,000000\,011011\,011011$.
e) $x = 111111\,000000\,100011\,100111$.
f) $x = 111111\,000000\,111000\,111000$.

18. Controlar que los coeficientes del enumerador de peso homogéneo del código de Golay en (3.7) son $a = 1$ y $b = -42$. Completar las cuentas para obtener la distribución de pesos de $\mathcal{G}_{24}$.
19. Demostrar el Teorema 3.9.
20. Encuentre todas las palabras código de $\mathcal{R}(1,3)$ y $\mathcal{R}(2,3)$ usando la definición. Controle que usando las matrices generadoras se obtienen los mismos códigos.
21. Calcule los códigos de Reed-Muller $\mathcal{R}(r,4)$, $0 \leq r \leq 4$.

Sobre códigos cíclicos.

1. Probar que los códigos $C_3(7)$ y $C_4(7)$ del Ejemplo 4.7 son equivalentes.
2. Encuentre todos los códigos cíclicos binarios de longitud 9 y liste las palabras código de cada uno de ellos.
3. Encontrar todos los códigos cíclicos ternarios de longitud 4. Escribir las matrices generadoras y de paridad para cada uno de ellos. Dar la lista de todas las palabras códigos de cada uno de ellos.
4. Describir el menor código cíclico que contiene la palabra 0011010.
5. ¿Es $\mathcal{H}_3(4)$ equivalente a un código cíclico?
6. Mostrar que el $[7,4]$-código binario $\langle\langle x^3+x+1\rangle\rangle$ y el $[7,3]$-código binario $\langle\langle x^4+x^3+x^2+1\rangle\rangle$ son códigos duales.
7. Probar que un código cíclico binario $C = \langle\langle g(x)\rangle\rangle$ contiene la palabra código $\mathbf{1}$ si y sólo si $g(\mathbf{1}) \neq 0$.
8. Probar la Proposición 4.10.
9. Utilice el truco de adivinar el dígito faltante en el código ISBN de un libro, para ganar apuestas en cafés literarios.

Referencias

[1] E. F. Assmus Jr, J.D. Key, *Designs and Their Codes*, Cambridge University Press, Cambridge Tracts in Mathematics **103**, (1992).
[2] P. J. Cameron, J.H. van Lint, *Designs, Graphs, Codes and their Links*, Cambridge University Press, London Mathematical Society Student Texts **22**, (1991).
[3] R. Hill, *A First Course in Coding Theory*, Clarendon Press, Oxford Applied Mathematics and Computing Sicence Series, (1986).
[4] D. G. Hoffman, D. A. Leonard, C. C. Lindner, K. T. Phelps, C. A. Rodgers, J. R. Wall, *Coding Theory. The essentials*, Pure and Applied Mathematics, Marcel Dekker, (1992).
[5] R. McEliece, *The Theory of Information and Coding Thoery*, Cambridge University Press, Encyclopedia of Mathematics and its Applications, (1977).
[6] V. Pless, *Introduction to the Theory of Error-Correcting Codes*, John Wiley & Sons, Wiley Series in discrete Mathematics and Optimization, (1998).
[7] O. Pretzel, *Error-Correcting Codes and Finite Fields*, Clarendon Press, Oxford Applied Mathematics and Computing Science Series, (1992).
[8] S. Roman, *Coding and Information Theory*, Springer, Graduate Texts in Mathematics **134**, (1992).
[9] S. A. Stepanov, *Codes on Algebraic Curves*, Kluwer Academic, (1999).
[10] J. H. van Lint, *Introduction to Coding Theory*, Springer, Graduate Texts in Mathematics **86**, (1982).

FaMAF-CIEM, Universidad Nacional de Córdoba, (5000) Córdoba, Argentina.
E-mail address: podesta@mate.uncor.edu