Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Matemática

Una Alternativa a la Decodificación de los Códigos BCH

Presentado para optar al grado de Licenciatura en Matemática

Autor
Univ. Israel Juan Mamani Quispe

Tutor
Lic. Eugenio Castaños Calle

La Paz - Bolivia
2022

Dedicatoria

Dedico este Proyecto a:

> Dios quien supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar en los problemas que se presentaban, enseñándome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

y a

> Mi hermosa familia, Ana mi esposa quien fue mi apoyo permanente y la persona que nunca perdió la confianza en mi, a mis hermosos hijos Anahi y David quienes son mi inspiración de superación y la razón de mi felicidad.

Índice general

Capítulo 1

Introducción

Todos los canales de comunicación contienen ruido en algún grado, esta llamada interferencia es causada por varias fuentes tales como: canales vecinos, impulsos eléctricos, deterioro del equipo, etc. Este ruido puede interferir con la transmisión de datos. Así como el mantener una conversación en una habitación ruidosa se hace más difícil cuando más ruido haya, así también la transmisión de datos se hace más difícil cuanto más ruidoso es el canal de comunicación. Con el propósito de mantener una conversación en una habitación ruidosa, tendrías que elevar la voz o repetir lo que digas. Este segundo método es uno que nos concierne; necesitamos añadir algo de redundancia a la transmisión con el objetivo de que el receptor pueda reconstruir el mensaje. A continuación, veremos varios ejemplos de técnicas que pueden ser usadas; en cada caso, los símbolos en el mensaje original son reemplazados por palabras-código en cuya construcción se ha incluido alguna redundancia.

Ejemplo 1. (Códigos de repetición)

Considere un alfabeto $\{A, B, C, D\}$. Queremos enviar una carta a través de un canal ruidoso que tiene una probabilidad de error igual a $p = 0{,}1$. Si queremos enviar C, por ejemplo, habrá un $90\,\%$ de probabilidad de que el símbolo recibido es C. Esto deja también una gran probabilidad de error.

Ahora veamos que pasa si repetimos el mensaje tres veces, es decir que enviamos CCC. Supongamos que ocurre un error y que la palabra recibida es CBC. Tomaremos el símbolo que aparece con más frecuencia como el mensaje, en este caso C. La probabilidad de que el mensaje correcto sea encontrado es la probabilidad de que las tres letras sean correctas más la probabilidad de que exactamente una de las tres letras esta equivocada:

$$(0{,}9)^3 + 3(0{,}9)^2(0{,}1) = 0{,}972$$

lo que deja una significativamente pequeña probabilidad de error.

Dos de los más importantes conceptos para los códigos son la detección de errores y la corrección de errores. Si hay a lo más dos errores, la repetición del código nos permite detectar el error que haya ocurrido. Si el mensaje recibido es CBC, entonces este puede ser un error de CCC o un error de BBB: y no podemos decir cual. Si a lo más un error se ha filtrado en el mensaje, entonces podemos corregir el error y deducir que el mensaje fue CCC. Notemos que si usamos dos repeticiones en lugar de tres, podemos detectar la existencia de un error, pero no podemos corregirlo. ¿Dime, CB viene de BB o CC?

Este ejemplo fue elegido para mostrar que los códigos correctores de errores pueden usar conjuntos de símbolos arbitrarios. Sin embargo, típicamente los símbolos que usaremos serán números matemáticos tales como enteros en modulo primo o cadenas binarias. Por ejemplo, podemos reemplazar las letras A, B, C, D por los números binarios (cadenas de 2 bits): 00, 01, 10, 11. El precedente proceso de repetición nos dará las palabras-código

$$000000, 010101, 101010, 111111.$$

Ejemplo 2. (Comprobación de paridad)

Supongamos que queremos enviar un mensaje de 7 bits. Y añadimos un octavo bit tal que la cantidad de bits no nulos es par. Por ejemplo, el mensaje 0110010 se convertiría en 0110010 y el mensaje 1100110 se convertiría en 11001100. Un error de un bit durante la transmisión seria inmediatamente descubierto pues el mensaje recibido tendría una cantidad impar de bits no nulos. Sin embargo, es imposible decir cual es el bit incorrecto, puesto que un error en cualquier bit produciría una cantidad impar de bits no nulos. Cuando un error es detectado, lo mejor que se puede hacer es volver a enviar el mensaje.

Ejemplo 3. (Código de paridad bidimensional)

La comprobación de paridad del ejemplo previo puede ser usado para diseñar un código que pueda corregir errores en un bit. El código de paridad bidimensional dispone los datos en un arreglo bidimensional para después añadir bits de paridad a lo largo de cada columna y cada fila.

Para ilustrar este código, supongamos que queremos codificar los 20 bits

$$10011011001100101011.$$

Distribuimos los bits en una matriz de 4×5

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

y calculamos los bits de paridad para cada fila y columna. Definimos el último bit en la esquina inferior derecha de la matriz extendida calculando la paridad de los bits de paridad que fueron calculados a lo largo de las columnas. Así obtenemos la matriz de 5×6

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & \mathbf{1} \\ 0 & 1 & 1 & 0 & 0 & \mathbf{0} \\ 1 & 1 & 0 & 0 & 1 & \mathbf{1} \\ 0 & 1 & 0 & 1 & 1 & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 1 \end{bmatrix}$$

Supongamos que esta matriz extendida de bits es transmitida y que se ocurre un error en el bit de la tercera fila y cuarta columna. El receptor acomoda los bits recibidos en una matriz de 5×6 y obtiene

$$\begin{matrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & \mathbf{1} & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{matrix}$$

Las paridades de la tercera fila y cuarta columna son impares, así esto nos dice que ha ocurrido un error en la intersección de la tercera fila y cuarta columna.

Si ocurriesen dos errores, este código este código podría detectar su existencia. Por ejemplo, si el error ocurriera en el segundo y tercer bit de la segunda fila, entonces la comprobación de paridad de la segunda y tercera columnas indicaría la existencia de dos bits errados. Sin embargo, en este caso no es posible el corregir los errores, pues existen varias posibles posiciones para ellos. Por ejemplo, si el segundo y tercer bit de la quinta fila fueran los incorrectos, entonces las comprobaciones de paridad serían las mismas que cuando los errores estaban en la segunda fila.

Ejemplo 4. (Código Hamming $[7, 4]$)

En el código Hamming el mensaje original consiste de bloques de cuatro dígitos binarios. Estos bloques son reemplazados por palabras código (que son bloques de siete bits) vía multiplicación a la derecha por la matriz

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

Por ejemplo, el mensaje 1100 es reemplazado por

$$\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \pmod 2$$

Como las cuatro primeras columnas de G son la matriz identidad, las primeras cuatro entradas de la palabra código son el mensaje original. Los restantes tres bits proveen la redundancia que permite la detección y corrección del error. En efecto, como bien veremos, podemos fácilmente corregir un error si este afecta solo uno de los siete bits en la palabra código.

Supongamos, por ejemplo, que la palabra código 1100011 es enviada pero es recibida como 1100001. ¿Como hacer para detectar y corregir el error? Escribamos G en la forma $[I_4, P]$, donde P es una matriz de 4×3. Formemos la matriz $H = [P^T, I_3]$, donde P^T es la transpuesta de P. Multipliquemos el mensaje recibido por la transpuesta de H:

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}^T \equiv \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\equiv \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \pmod 2$$

Esto es la sexta fila de H^T, lo que significa que existe un error en el sexto bit del mensaje recibido. Por tanto la palabra código correcta es 0001111, y los primeros cuatro dígitos dan el mensaje original. Si no hubiese error alguno el resultado de multiplicar

por H^T sería $(0,0,0)$ así quedaría reconocido que ninguna corrección es necesaria. Este misterioso procedimiento será explicado cuando discutamos los códigos de Hamming en el capitulo 5. Por el momento, notemos que este código nos permite corregir errores de un bit con bastante eficiencia.

El código Hamming $[7,4]$ brinda una significativa mejora sobre el código de repetición. En el código de Hamming, si queremos enviar cuatro bits de información, debemos transmitir 7 bits. Hasta dos errores pueden ser detectados y hasta un error puede ser corregido. Para que un código de repetición pueda alcanzar este nivel de detección y corrección de errores, necesitaría transmitir 12 bits con el propósito de enviar un mensaje de 4 bits. Después expresaremos esto matemáticamente diciendo que la razón-código del código de Hamming es $4/7$, mientras que la razón del código de repetición es $4/12 = 1/3$. Generalmente, mientras más grande sea la razón de un código es mejor, pero si es demasiado grande se pierde capacidad de corregir errores. Por ejemplo, enviando un mensaje de 4 bits como el mismo tiene una razón de código igual a 1 pero esto es insatisfactorio en la mayoría de las situaciones pues no existe capacidad alguna de corregir errores.

Ejemplo 5. (Código ISBN)

El Estándar Internacional para la Numeración de Libros (ISBN) nos proporciona otro ejemplo de código corrector de errores. El ISBN es un código de 10 dígitos que es asignado a cada libro cuando este es publicado. Por ejemplo, este libro tiene un numero ISBN 0-13-061814-4. El primer dígito representa el lenguaje que se ha usado; en este caso 0 indica inglés. Los próximos dos dígitos representan al publicador, el 13 esta asociado a la Prentice Hall. Los próximos seis numeros corresponden al numero de identidad del libro que es asignado por el publicador. El décimo dígito es elegido de tal manera que el numero ISBN $a_1, a_2, \ldots, a_{10}$ satisfaga la relación

$$\sum_{j=1}^{10} j a_j \equiv 0 \pmod{11}.$$

Notar que todos los cálculos se realizan en modulo 11. Los primeros nueve numeros $a_1, a_2, \ldots, a_9$ se toman de $\{0, 1, \ldots, 9\}$ pero a_{10} puede ser 10, en este caso es representado por el símbolo X. Suponiendo que el numero ISBN $a_1, a_2, \ldots, a_{10}$ es enviado a través de un canal ruidoso, o es escrito sobre un formulario para ordenar un libro, y es recibido como $x_1, x_2, \ldots, x_{10}$. El código ISBN puede ordenar un simple error, o un doble error debido a la transposición de dos dígitos. Para determinar esto, el receptor debe calcular la suma ponderada

$$S = \sum_{j=1}^{10} j x_j \pmod{11}.$$

Si $S \equiv 0 \pmod{11}$, entonces no se han detectado errores, aunque existe una pequeña posibilidad que haya un error pero sea indetectable. Por otro lado, cuando detectemos un error, quizá no podamos corregirlo.

Si $x_1, x_2, \ldots, x_{10}$ es el mismo que $a_1, a_2, \ldots, a_9$ excepto en un lugar x_k, podemos escribir $x_k = a_k + e$ donde $e \neq 0$. Calculando S da

$$S = \sum_{j=1}^{10} j a_j + ke \equiv ke \pmod{11}$$

así, si un simple error ocurre podemos detectarlo. El otro tipo de error que puede ser formalmente detectado es cuando a_k y a_l han sido transpuestos. Este es uno de los mas comunes errores que ocurren cuando un numero es copiado. En este caso $x_l = a_k$ y $x_k = a_l$. Calculando S da

$$\begin{aligned} S = \sum_{j=1}^{10} j x_j &= \sum_{j=1}^{10} j a_j + (k-l)a_l + (l-k)a_k \pmod{11} \\ &\equiv (k-l)(a_l - a_k) \pmod{11}. \end{aligned}$$

Si $a_l \neq a_k$, entonces la suma no es igual a 0, y un error ha sido detectado.

Ejemplo 6. (Código de Hadamard)

Este código fue usado por la nave espacial Mariner en 1969 para enviar fotografías a la Tierra y tiene 64 palabras código; 32 son representadas por las filas de la matriz de 32×32

$$\begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 1 & -1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & -1 & -1 & 1 & \cdots & -1 \end{bmatrix}.$$

La matriz es construida como sigue. Las filas y las columnas se numeran de 0 a 31. Para obtener la entrada h_{ij} de la i-ésima fila y j-ésima columna, se escriben $i = a_4a_3a_2a_1a_0$ y $j = b_4b_3b_2b_1b_0$ en binario. Entonces

$$h_{ij} = (-1)^{a_0b_0 + a_1b_1 + \cdots + a_4b_4}.$$

Por ejemplo, cuando $i = 31$ y $j = 3$, tenemos $i = 11111$ y $j = 00011$. Por lo tanto, $h_{ij} = (-1)^2 = 1$. Las otras 32 palabras código son obtenidas usando las filas de $-H$. Note que el producto punto de cualesquiera dos filas de H es 0, salvo que las dos filas sean iguales, y en tal caso el producto punto da 32. Cuando el Mariner envía una fotografía, cada pixel tiene una oscuridad dada por un numero de 6 bits, este es cambiado por una de las palabras código y transmitido. Un mensaje recibido (esto es, una cadena de 1s y -1s de longitud 32) puede ser decodificado (esto es corregido para que vuelva a ser una palabra código) como sigue. Tomando el producto punto del mensaje con cada fila de H. Si el mensaje es correcto, se tendrá un producto punto nulo con todas las filas salvo con una que dará ± 32. Si el producto punto es 32, la palabra código es esa fila de H; y si es -32, la palabra código es la correspondiente fila de $-H$. Si el mensaje tiene un error, todos los productos punto serán ± 2, salvo uno que dará ± 30, y que también dará la fila correcta de H o $-H$. Si hay dos errores, todos los productos punto serán $0, \pm 2, \pm 4$, salvo uno que será $\pm 32, \pm 30$ o ± 28. Continuando, veremos que si hay siete errores, todos los productos punto estarán entre -14 y -14, excepto uno que estará entre -30 y -16 o entre 16 y 30, que producirá la correcta palabra-código. Con ocho o más errores, los productos punto empiezan al solaparse, haciendo que la corrección sea imposible. Sin embargo, la corrección es posible hasta 15 errores, pues bastan 16 errores para cambiar una palabra-código en otra.

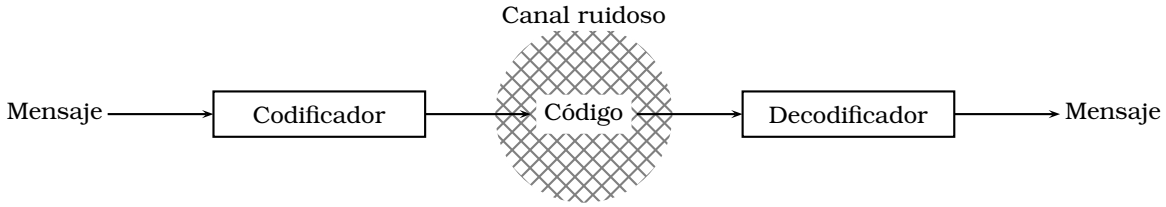
Este código tiene una relativamente baja razón de 6/32, puesto que usa 32 bits para enviar un mensaje de 6 bits. Sin embargo, esto es balanceado por una elevada razón de corrección de errores. Como los mensajes del Mariner eran medianamente débiles, el potencial de errores era alto, así que una elevada capacidad para la corrección

de errores era necesaria. La otra opción era el incrementar la potencia de la señal y emplear un código con elevada razón y menor capacidad para corregir errores. La transmisión habría tomado menos tiempo y potencialmente también se hubiese usado menos energía. Sin embargo, en este caso, resulto que usar una señal débil compensaba sobradamente la perdida en velocidad. Este asunto (técnicamente conocido como *ganancia de código*) es de importante consideración en la elección de que código usar en una aplicación dada.

Capítulo 2

Códigos correctores de errores

Un transmisor comienza con un mensaje y lo codifica para obtener palabras-código consistentes de secuencias de símbolos. Estos son transmitidos a través de un canal ruidoso (representado en la siguiente figura) para el receptor.



Esquema de codificado y decodificado.

Muchas veces las secuencias de símbolos que son recibidas contienen errores y por tanto potencialmente no serán palabras-código. El receptor debe decodificar, es decir, corregir los errores para que los símbolos recibidos vuelvan a ser palabras-código y así recobrar el mensaje original.

Los símbolos usados para construir las palabras-código pertenecen a un alfabeto. Cuando el código consiste de los dígitos binarios 0 y 1, el código es llamado un *código binario*. Un código que usa secuencias de tres símbolos, a menudo representados como enteros $\mod 3$, es llamado *código ternario*. En general, un código que usa un alfabeto consistente de q símbolos es llamado *código* $\mathbf{q}$*-ario*.

DEFINICIÓN. Si $\mathcal{A}$ es un alfabeto y $\mathcal{A}^n$ denota el conjunto de las n-tuplas de elementos de $\mathcal{A}$. Un *código de longitud* $\mathbf{n}$ es un subconjunto no vació de $\mathcal{A}^n$.

Las n-tuplas que constituyen un código son las que hasta ahora hemos llamado *palabras-código* (o vectores-código), nombre que conservaremos. Por ejemplo, en un código binario de repetición donde cada símbolo es repetido tres veces, el alfabeto es el conjunto $\mathcal{A} = \{0, 1\}$ y el código es el conjunto $\{(0, 0, 0), (1, 1, 1)\} \subset \mathcal{A}^3$.

Hablando estrictamente, los códigos en la definición son llamados *códigos en bloque*. Existen otros códigos donde las palabras-código pueden tener varias longitudes, pero estos no son de nuestro interés, por lo que nos enfocaremos exclusivamente en los códigos en bloque.

Para un código que es un subconjunto aleatorio de $\mathcal{A}^n$, decodificar puede ser un proceso que consuma mucho tiempo, por esto los códigos más útiles son los subconjuntos de $\mathcal{A}^n$ que satisfacen condiciones adicionales. Lo más común es requerir que $\mathcal{A}$ sea un campo finito, tal que $\mathcal{A}^n$ sea un espacio vectorial, y que el código sea un subespacio de este espacio vectorial. Tales códigos son llamados lineales y serán discutidos más adelante.

Por el resto de este capitulo, sin embargo, trabajaremos con códigos arbitrarios, posiblemente no lineales. Siempre asumiremos que nuestras palabras-código son vectores n-dimensionales.

Con el propósito de decodificar, será útil el adoptar una forma de medir cuan cercanos están dos vectores uno de otro. Esto se puede realizar a través de la distancia de Hamming.

DEFINICIÓN. Sean x, y dos vectores en $\mathcal{A}^n$, es decir $x = x_1x_2 \dots x_n$, $y = y_1y_2 \dots y_n$ la *distancia de Hamming* entre los vectores x y y, denotada por $d(u, v)$ es el número de subíndices i tales que $x_i \neq y_i$ (o lugares donde los dos vectores difieren).

Por ejemplo, si usamos vectores binarios y usamos los vectores $x = 10101010$ y $y = 10111000$, entonces x y y difieren en dos lugares (el cuarto y el séptimo) así $d(x, y) = 2$.

Como otro ejemplo, supongamos que trabajamos con el alfabeto español, entonces $d(cuarta, puerco) = 4$ pues las dos cadenas difieren en cuatro lugares.

La importancia de la distancia de Hamming $d(u, v)$ es que mide el mínimo número de errores que se necesitan para convertir u en v. A continuación se dan algunas de sus propiedades básicas.

Teorema 1. La distancia de Hamming $d(u, v)$ es una métrica sobre $\mathcal{A}^n$, esto significa que satisface

1. $d(u, v) \geq 0$, y $d(u, v) = 0$ si y solo si $u = v$.
2. $d(u, v) = d(v, u)$ para todo u, v.
3. $d(u, v) \leq d(u, w) + d(w, v)$ para todo u, v, w.

La tercera propiedad es a menudo llamada, la desigualdad triangular.

Demostración. (1) $d(u, v) = 0$ es exactamente lo mismo que decir que u y v difieren en ningún lugar, es decir que $u = v$. La parte (2) es obvia. Para la parte (3) observe que si u y v difieren en un lugar, entonces o u y w difieren en un lugar o w y v difieren en un lugar, o ambos. Por tanto, el número de lugares en los cuales u y v difieren es menor que o igual al número de lugares en los que difieren u y w, mas el número de lugares en los que difieren w y v. □

Para un código C, uno puede calcular la distancia de Hamming entre cualesquiera dos palabras-código. En esta tabla de distancias hay un mínimo valor que por su importancia definimos a continuación.

DEFINICIÓN. La distancia mínima de un código C, denotada por $d(C)$, es

$$d(C) = \text{mín}\{d(x,y)|\, x,y \in C, x \neq y\}.$$

la más pequeña distancia entre cualesquiera dos palabras-código diferentes en C.

La mínima distancia de C es un número muy importante, pues este da el mínimo numero de errores necesarios para convertir una palabra-código en otra.

Cuando una palabra-código es transmitida a través de un canal ruidoso, errores son introducidos en algunas de las entradas del vector. Corregiremos estos errores, encontrando la palabra código cuya distancia de Hamming del vector recibido es la más pequeña posible. En otras palabras, cambiaremos el vector recibido por una palabra-código cambiando la menor cantidad de lugares posibles. Esto es llamado *decodificar por el más cercano vecino*.

Diremos que el código puede *detectar* hasta s errores si cambiando una palabra-código en a lo más s lugares no puede convertirse en otra palabra-código. El código puede *corregir* hasta t errores si, cada vez que cambiamos t o menos lugares en una palabra-código c, entonces la más cercana palabra-código aún es c. Esta definición no nos dice nada sobre un algoritmo eficiente para corregir los errores. Simplemente requiere que el decodificar por el más cercano vecino de la respuesta correcta cuando haya a lo más t errores. Un importante resultado de la teoría de códigos correctores de errores es el siguiente.

Teorema 2. 1. Un código puede detectar hasta s errores si $d(C) \geq s+1$. 2. Un código C puede corregir hasta t errores si $d(C) \geq 2t+1$.

Demostración. (1) Suponga que $d(C) \geq s+1$. Si la palabra-código c es enviada y ocurren s o menos errores, entonces el mensaje recibido r no puede ser una palabra-código diferente. Por lo tanto un error ha sido detectado. (2) Supongamos que $d(C) \geq 2t+1$. Y asumamos que la palabra-código c es enviada y la palabra-código recibida r tiene t o menos errores, es decir $d(c,r) \leq t$. Si c_1 es cualquier otra palabra-código además de c, afirmamos que $d(c_1,r) \geq t+1$. Para ver esto, supongamos que $d(c_1,r) < t$. Entonces al aplicar la desigualdad, tenemos

$$2t+1 \leq d(C) \leq d(c,c_1) \leq d(c,r) + d(c_1,r) \leq t+t = 2t.$$

Esta es una contradicción, así $d(c_1,r) \geq t+1$. Como r tiene t o menos errores la decodificación por el más cercano vecino sucesivamente decodifica r en c □

¿Cómo hace uno para encontrar al más cercano vecino? Una manera es calcular la distancia entre el mensaje recibido r y cada una de las palabras código, entonces selecciona la palabra código con la más pequeña distancia de Hamming. En la practica, esto es impracticable para grandes códigos. En general, el problema de decodificar es desafiante y es considerable el esfuerzo que se ha dedicado a buscar algoritmos decodificadores más veloces. En los próximos capítulos, discutiremos una pocas técnicas para decodificar que han sido desarrolladas para clases especiales de códigos.

Antes de continuar, es conveniente introducir algo de notación.

Notación. Un código de longitud n, con M palabras-código, y con distancia mínima $d = d(C)$, será llamado un *(n,M,d)-código*.

Cuando tratemos con códigos lineales, emplearemos una notación similar, los llamaremos [n,k,d]-código. Notese que esta última notación emplea corchetes mientras que la precedente usa paréntesis.

El código binario de repetición $\{(0,0,0),(1,1,1)\}$ es un (3,2,3)-código. El código de Hadamard es un (32,64,16)-código (que puede corregir hasta 7 errores por que $16 \geq 2 \cdot 7 + 1$).

Si tenemos un (n, M, d)-código q-ario, entonces definimos la *razón de código*, o *tasa de información*, R por

$$R = \frac{\log_q M}{n}.$$

Por ejemplo, para el código de Hadamard, $R = \log_2(64)/32 = 6/32$. La razón de código representa la proporción de símbolos en los datos de entrada a el número de símbolos del código transmitido. Este es un importante parámetro a considerar cuando se implementan sistemas en el mundo real, pues es cuando representa que fracción del ancho de banda se esta usando para transmitir los datos. La razón de código ya fue mencionada en los ejemplos 4 y 6 del capitulo 1. Una pocas limitaciones sobre la razón de código se discutirán en el capitulo 3.

Dado un código es posible construir otros códigos que son esencialmente el mismo. Supongamos que tenemos una palabra-código c que es expresada como $c = c_1 c_2 \cdots c_n$. Entonces podemos definir una permutación posicional de c permutando el orden de las entradas en c. Por ejemplo, el nuevo vector $c' = c_2 c_3 c_1$ es una permutación posicional de $c = c_1 c_2 c_3$. Otro tipo de operación que se puede hacer es una permutación de los símbolos. Supongamos que tenemos una permutación de los símbolos q-arios. Entonces podemos fijar una posición y aplicar esta permutación de los símbolos a la posición fijada de cada palabra-código. Por ejemplo, supongamos que tenemos la siguiente permutación de los símbolos ternarios $\{0 \to 2, 1 \to 0, 2 \to 1\}$,y que tenemos las siguientes palabras-código: 012, 021 y 201. Entonces aplicando la permutación a la segunda posición de todas las palabras-código obtenemos los siguientes vectores: 002, 011 y 221.

Formalmente diremos que dos códigos son *equivalentes* si un código puede ser obtenido a partir del otro por una serie de las siguientes operaciones:

1. Permutando las posiciones del código.
2. Permutando los símbolos que aparecen en una posición determinada en todas las palabras-código.

Es fácil ver que todos los códigos equivalentes a un (n, M, d)-código son también (n, M, d)-códigos. Sin embargo, para ciertas elecciones de n, M y d puede haber varios (n, M, d)-código equivalentes.

Capítulo 3

Cotas sobre códigos

Hemos mostrado que un (n, M, d)-código puede corregir t errores si $d \geq 2t + 1$. Por lo tanto, nos gustaría que la distancia mínima d sea tan grande que podamos corregir tantos errores como sea posible. Pero también nos gustaría que M sea tan grande que la razón de código R se acerque a 1 tanto como sea posible. Esto nos llevaría a usar eficientemente el ancho de banda a la hora de transmitir mensajes a través de canales ruidosos. Desafortunadamente, cuando d decrece hace que n crezca o que M decrezca.

En este capitulo, estudiaremos las restricciones sobre n, M y d sin preocuparnos por los aspectos prácticos tales como si los códigos con buenos parámetros tienen eficientes algoritmos decodificadores. Estos resultados aun serán útiles pues nos darán una idea de cuan bueno es un código comparado a los limites teóricos.

Primero, conseguiremos cotas superiores para M en términos de n y d. Entonces mostraremos que existen códigos con M mayor que ciertas cotas inferiores. Finalmente, veremos como nuestros ejemplos se comparan con estas cotas.

Cotas superiores

Nuestro primer resultado fue dado por R. Singleton en 1964 y es conocido como la *cota de Singleton*.

Teorema 3. Si C es un (n, M, d)-código q-ario, entonces $M \leq q^{n-d+1}$.

Demostración. Para una palabra-código $c = (a_1, \ldots, a_n)$, sea $c' = (a_d, \ldots, a_n)$. Si $c_1 \neq c_2$ son dos palabras-código, entonces ellas difieren en al menos d lugares. Como c_1' y c_2' son obtenidas removiendo $d-1$ entradas de c_1 y c_2, ellos deben diferir en al menos un lugar, así $c_1' \neq c_2'$. Por lo tanto, el número M de palabras código c es igual al número de vectores c' obtenidos de esta manera. Existen a lo más q^{n-d+1} vectores c' pues hay $n-d+1$ posiciones en estos vectores. Esto implica que $M \leq q^{n-d+1}$, como se quería. □

COROLARIO 4. La razón de código de un (n, M, d)-código q-ario es a lo más $1 - \frac{d-1}{n}$.

Demostración. El corolario se sigue inmediatamente de la definición de razón para un código. □

El corolario implica que si la distancia mínima relativa d/n es grande, la razón de código es forzada a ser pequeña.

Un código que satisface la igualdad de la cota de Singleton es llamado *código MDS* (Máxima distancia separable). La cota de Singleton puede reescribirse como $q^d \leq q^{n+1}/M$, así un código MDS tiene el más grande valor posible de d para valores dados de n y M. Los códigos de Reed-Solomon son un importante clase de código MDS.

Antes de derivar otra cota superior, necesitamos introducir una interpretación geométrica que es útil en la corrección de errores. Una esfera de Hamming de radio t centrada en la palabra-código c es denotada por $B(c,t)$ y esta formada por todos los vectores que están a lo más una distancia de Hamming t de la palabra-código c. Esto es, un vector pertenece a la esfera de Hamming $B(c,t)$ si $d(c,u) \leq t$. Calcularemos el número de vectores en $B(c,t)$ en el siguiente lema.

LEMA 5. Una esfera $B(c,t)$ en un espacio q-ario n-dimensional tiene

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{r}(q-1)^r$$

elementos.

Demostración. Primero calcularemos el número de vectores que están a una distancia 1 de c. Estos vectores son los que difieren de c en exactamente una posición. Hay n posibles locaciones y $q-1$ maneras para hacer que la entrada difiera. De esta manera el número de vectores que están a una distancia de Hamming 1 de c es $n(q-1)$. Ahora calcularemos el número de vectores que están a una distancia m de c. Existen $\binom{n}{m}$ maneras en las que se puede elegir m lugares de c para cambiarlos. Para cada una de estas m posiciones, existen $q-1$ elementos diferentes para elegir de los correspondientes símbolos de c. Por lo tanto, hay

$$\binom{n}{m}(q-1)^m$$

vectores que están a una distancia de Hamming m de c. Así tenemos el resultado

$$\binom{n}{0} + \binom{n}{1}(q-1) + \binom{n}{2}(q-1)^2 + \cdots + \binom{n}{r}(q-1)^r$$

donde se ha incluido al mismo vector c, usando la identidad $\binom{n}{0} = 1$. □

Ahora podemos establecer la *cota de Hamming*, que es también llamada la cota de la esfera empaquetada.

Teorema 6. Si C es un (n, M, d)-código q-ario con $d \geq 2t+1$. Entonces

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

Demostración. Al rededor de cada palabra-código c colocamos una esfera de Hamming de radio t. Como la distancia mínima del código es $d \geq 2t+1$, estas esferas no se solaparan. El número total de vectores en todas las esferas de Hamming no puede ser mayor a q^n. Así tenemos que (el numero de palabras-código)$\times$(número de elementos por esfera) será igual a

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

Esto produce la desigualdad deseada para M. □

Un (n, M, d)-código con $d = 2t+1$ que satisfaga la igualdad de la cota de Hamming es llamado un *código perfecto*. Un código perfecto corrector de t errores es uno tal que las M esferas de Hamming de radio t con centro en las palabras-código cubren el espacio entero de las n-tuplas q-arias. Los códigos de Hamming (capitulo 5) y el código de Golay G_{23} son perfectos. Otros ejemplos de códigos perfectos son el trivial $(n, q^n, 1)$-código obtenido al tomar todas las n-tuplas, y los códigos binarios de repetición de longitud impar.

Los códigos perfectos han sido objeto de mucho estudio, y son interesantes desde muchos puntos de vista. La lista completa de códigos perfectos es conocida, e incluyen los precedentes ejemplos, a los que se deben añadir $(11, 6, 5)$-código ternario construido por Golay. Dejaremos al lector una advertencia. Con un nombre como el de código perfecto, uno puede asumir que estos códigos son los mejores códigos correctores de errores. Esto sin embargo, no es verdad, hay códigos correctores de errores, tales como los códigos Reed-Solomon, que no son códigos perfectos pero aun así tienen una mejor capacidad de corregir errores en ciertas situaciones que los códigos perfectos.

Cotas inferiores

Uno de los problemas centrales de la teoría de códigos correctores de errores es el encontrar el código más grande de una longitud dada y distancia mínima d dada. Esto nos lleva a la siguiente definición

> *DEFINICIÓN.* Si el alfabeto $\mathcal{A}$ tiene q elementos. Dados n y d con $d \leq n$, el mayor valor de M tal que un (n, M, d)-código exista es denotado por $A_q(n, d)$.

Siempre podemos encontrar al menos un (n, M, d)-código: Fijando un elemento a_0 de $\mathcal{A}$. Sea C el conjunto de todos los vectores $(a, a, \ldots, a, a_0, \ldots, a_0)$ (con d copias de a y $n-d$ copias de a_0) con $a \in \mathcal{A}$. Hay q de tales vectores, y ellos están a una distancia d uno de otro, así tenemos un (n, q, d)-código. Esto da la trivial cota inferior $A_q(n, d) \geq q$. Obtendremos mejores cotas después.

Es fácil ver que $A_q(n, 1) = q^n$: cuando los códigos tienen mínima distancia $d = 1$ podemos tomar a todas las n-tuplas q-arias como el código. Y al otro extremo, $A_q(n, n) = q$.

La siguiente cota inferior, conocida como la cota Gilbert-Vashamov, fue descubierta en la década de 1950.

Teorema 7. Dados n, d con $n \geq d$, existe un (n, M, d)-código q-ario con

$$M \geq \frac{q^n}{\sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j} = G.$$

esto significa que $A_q(n, d) \geq G$.

Demostración. Comencemos con un vector c_1 y removamos todos los vectores en $\mathcal{A}^n$ (donde $\mathcal{A}$ es un alfabeto con q símbolos) que están en una esfera de Hamming de radio $d-1$ sobre este vector. Ahora elijamos otro vector c_2 de los restantes. Como todos los vectores con distancia a lo más $d-1$ de c_1 han sido removidos, $d(c_1, c_2) \geq d$. Ahora removamos todos los vectores que están a una distancia a lo más $d-1$ de c_2, y elijamos c_3 de entre los vectores restantes. No se puede tener $d(c_3, c_1) \leq d-1$ ni $d(c_3, c_2) \leq d-1$, pues todos los vectores que satisfacen estas desigualdades han sido removidos. Por lo tanto $d(c_3, c_i) \geq d$ para $i = 1, 2$. Continuando se esta manera podemos elegir $c_4, c_5, \ldots$, hasta que no haya más vectores.

La selección de un vector remueve a lo más

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

vectores del espacio. Si hemos elegido M vectores $c_1, \ldots, c_M$, entonces hemos removido a lo más

$$M \sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j$$

vectores, por el precedente lema. Podemos continuar hasta que todos los q^n vectores hayan sido removidos, lo que significa que podemos continuar al menos hasta que

$$M \sum_{j=0}^{d-1} \binom{n}{j}(q-1)^j \geq q^n.$$

Por lo tanto, existe un código $\{c_1, \ldots, c_M\}$ con M satisfaciendo la precedente desigualdad.

Como $A_q(n, d)$ es el más grande de los M, este también satisface la desigualdad.

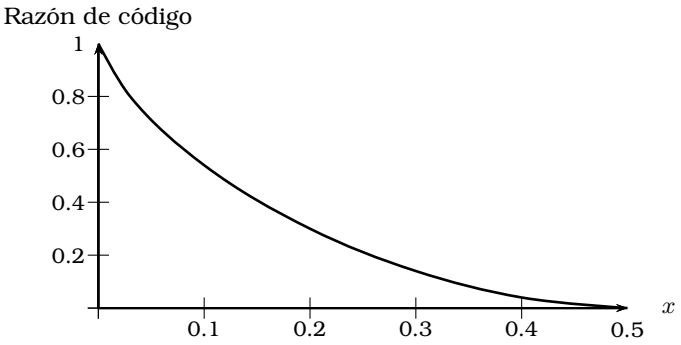
Aun queda un pequeño tecnicismo que debe mencionarse. Hemos construido un (n, M, e)-código con $e \geq d$. Sin embargo, modificando un poco las entradas en c_2 si es necesario, podemos hacer que $d(c_2, c_1) = d$. Los restantes vectores serán entonces elegidos por el anterior procedimiento. Esto produce un código donde la distancia mínima es exactamente d. □

Si queremos enviar palabras-código con n bits a través de un canal ruidoso, y hay una probabilidad p de que cualquier bit dado sea corrompido, entonces podemos esperar que el número de errores sea aproximadamente pn donde n es grande. Por lo tanto necesitamos un (n, M, d)-código con $d > 2pn$. Por lo tanto necesitamos considerar (n, M, d)-códigos con $d/n \approx x > 0$, para algún $x > 0$ dado. ¿Como afectara esto a M y a la razón del código?

Aquí esta lo que sucede. Fijando q y eligiendo x con $0 < x < 1 - 1/q$. La asintótica cota Gilbert-Vashamov dice que existe una secuencia de (n, M, d)-códigos q-arios con $n \to \infty$ y $d/n \to x$ tales que la razón de código se aproxima a un limite $\geq H_q(x)$, donde

$$H_q(x) = 1 - x\log_q(q-1) + x\log_q(x) + (1+x)\log_q(1-x).$$

La gráfica de $H_2(x)$ se muestra a continuación



Gráfica de $H_2(x)$.

Esta claro que queremos códigos con una alta capacidad de corregir errores (esto es, con x grande), y con una elevada razón ($= k/n$). El resultado asintótico dice que existen códigos bastante buenos (es decir, con una buena razón de código y una buena capacidad de corregir errores) y arbitrariamente cercanos a (o sobre) la gráfica.

La existencia de ciertas secuencias de códigos teniendo un limite de razón de código estrictamente mayor que $H_q(x)$ (para ciertos x y q) fue probado en 1982 por Tsfasman, Vladut y Zink usando códigos de Goppa provenientes de la geometría algebraica.

Ejemplo 7. Considera el código binario de repetición C de longitud 3 con los dos vectores $(0,0,0)$ y $(1,1,1)$. Este es un $(3,2,3)$-código. La cota de Singleton dice que $2 = M \leq 2$, así C es un código MDS. La cota de Hamming dice que

$$2 = M \leq \frac{2^3}{\binom{3}{0} + \binom{3}{1}} = 2,$$

así que C también es perfecto. La cota Gilbert-Vashamov dice que existe un $(3, M, 3)$-código binario con

$$M \geq \frac{2^3}{\binom{3}{0} + \binom{3}{1} + \binom{3}{2}} = \frac{8}{7},$$

es decir que $M \geq 2$.

El código Hamming $[7, 4]$ tiene $M = 16$ y $d = 3$, así que este es un $(7, 16, 3)$-código. La cota de Singleton dice que $16 = M \leq 2^4$, por lo que es un código MDS. La cota de Hamming dice que

$$16 = M \leq \frac{2^7}{\binom{7}{0} + \binom{7}{1}} = 16,$$

así que el código es perfecto. La cota Gilbert-Vashamov dice que existe un $(7, M, 3)$-código con

$$M \geq \frac{2^7}{\binom{7}{0} + \binom{7}{1} + \binom{7}{2}} = \frac{128}{29} \approx 4{,}4,$$

así que el código de Hamming es mucho mejor que esta cota inferior. Códigos que tienen algoritmos correctores de errores eficientes y que también exceden la cota Gilbert-Vashamov son relativamente raros.

El código de Hadamard es un $(32, 64, 16)$-código binario. La cota de Singleton dice que $64 = M \leq 2^{17}$, así que esto no es muy claro en este caso. La cota de Hamming dice que

$$64 = M \leq \frac{2^{32}}{\sum_{j=0}^{7} \binom{32}{j}} \approx 951{,}3.$$

La cota Gilbert-Vashamov dice que existe un $(32, M, 16)$-código binario con

$$M \geq \frac{2^{32}}{\sum_{j=0}^{15} \binom{32}{j}} \approx 2{,}3.$$

Capítulo 4

Códigos lineales

Cuando tu tienes una conversación con un amigo por medio de un teléfono celular tu voz se convierte en un dato digital al que se le ha aplicado un código corrector de errores antes de ser enviado. Cuando tu amigo recibe los datos, los errores en la transmisión deben ser contabilizados para ser decodificados por el código corrector de errores. Solo después de que los datos han sido decodificados estos se convierten en los sonidos que representan tu voz.

El tiempo que toma el decodificar un paquete de datos es critico en tal aplicación. Si esto toma varios segundos, entonces la tardanza se hará irritante y el sostener la conversación será difícil.

El problema de decodificar eficientemente un código es por tanto de critica importancia. Con el propósito de decodificar rápidamente, será útil el tener una manera más estructurada de escoger el código, mejor que tomar un subconjunto aleatorio de $\mathcal{A}^n$. Esta es una de las razones principales para el estudio de los códigos lineales. De aquí en adelante nos restringiremos al estudio de los códigos lineales.

De aquí en adelante, el alfabeto $\mathcal{A}$ será un campo finito F. Para mucho de lo que haremos, el lector puede asumir que F es $\mathbb{Z}_2 = \{0, 1\}$ =los enteros modulo 2, en tal caso trabajaremos con vectores binarios. Otro ejemplo concreto de un campo finito es $\mathbb{Z}_p$ =los enteros modulo p primo.

El conjunto de vectores n-dimensionales con entradas en F es denotado por F^n. Ellos forman un espacio vectorial sobre F. Recordemos que un subespacio de F^n es un subconjunto no vació S que es cerrado bajo las combinaciones lineales, esto quiere decir que si $s_1, s_2 \in S$ y $a_1, a_2 \in F$, entonces $a_1 s_1 + a_2 s_2 \in S$.

DEFINICIÓN. Sea F un campo finito, n un entero positivo y C un subespacio del espacio vectorial $V = F^n$. Entonces se dirá que C es un *código lineal* sobre F. Si C es un subespacio de dimensión k, entonces diremos que C es un $[n, k]$-código. Además si el código C tiene mínima distancia d, entonces diremos que C es un $[n, k, d]$-código.

Cuando $F = \mathbb{Z}_2$, la definición es más sencilla. Un código binario de longitud n y dimensión k es el conjunto de 2^k n-tuplas binarias (las palabras-código) tales que la suma de cualesquiera dos palabras-código es siempre una palabra-código.

Muchos de los códigos que hemos visto son códigos lineales. Por ejemplo, el código

binario de repetición $\{000, 111\}$ es un $[3, 1, 3]$-código lineal sobre el campo $F = \{0, 1\}$.

El código de comprobación de paridad es un $[8, 7]$-código lineal. Este consiste de vectores binarios de longitud 8 tales que la suma de las entradas da 0 en modulo 2. No es difícil el mostrar que el conjunto de tales vectores forma un subespacio. Los vectores

$$10000001, 01000001, \ldots, 00000011$$

forma una base de este subespacio. Como existen siete vectores base, es subespacio es 7-dimensional.

El código Hamming $[7, 4]$ es un $[7, 4]$-código lineal. Cada palabra-código es una combinación lineal de las cuatro filas de la matriz G. Como estas cuatro filas generan el código y son linealmente independientes, ellos forman una base.

El código binario $\{000, 100, 111\}$ no es lineal porque $100 + 111$ es igual a 011 que no es una palabra código.

El código ISBN no es lineal. Este consiste de un conjunto de vectores 10-dimensionales con entradas en $\mathbb{Z}_{11}$. Sin embargo, este no es cerrado bajo las combinaciones lineales

Se debe observar la diferencia entre la notación (n, M, d) para un código de M palabras usado en el capitulo 2 y la notación (n, k, d) para un código lineal de dimensión k.

Si F es un campo que tiene q elementos y C es un $[n, k]$-código lineal sobre F. Entonces siendo un subespacio de F^n de dimensión k, C tiene una base de k elementos, denotada por $b_1, \ldots, b_k$. Así, cada elemento $x \in C$ puede escribirse de forma única como combinación lineal de los elementos de la base; esto es,

$$x = \alpha_1 b_1 + \cdots + \alpha_k b_k,$$

donde $\alpha_1, \ldots, \alpha_k \in F$. Como cada α_i se puede escoger de q maneras, se sigue que C tiene q^k elementos. Por lo tanto, concluimos que un $[n, k]$-código lineal q-ario esta formado por q^k palabras-código. En particular un $[n, k]$-código lineal binario esta formado por 2^k palabras-código.

Con miras en el desarrollo posterior, F denotara un campo finito. Un campo de q elementos será denotado por $\mathbb{F}_q$ (recordemos que el numero de elementos en un campo finito es una potencia de algún primo). Habitualmente los elementos de F^n son vectores fila, pero dependiendo del contexto, un elemento $x \in F$ lo escribiremos como $(x_1, \ldots, x_n)$ o $[x_1, \ldots, x_n]$ o simplemente $x_1 \ldots x_n)$ como hasta ahora. También escribiremos x^T par denotar la transpuesta de x. El símbolo 0 denotara al elemento cero en F así como al vector cero $(0, 0, \ldots, 0) \in F^n$.

Matrices generadora y de paridad

Existen dos matrices asociadas a un código lineal que juegan un importante papel en la teoría de códigos. Una es la matriz generadora y la otra es la matriz de paridad, con las que trataremos en adelante.

DEFINICIÓN. Si C es un $[n, k]$-código lineal, y G es una matriz de tamaño $k \times n$ cuyas filas forman una base de C, entonces nos referiremos a G como la matriz generadora del código C.

Un $[n,k]$-código lineal es completamente determinado por una matriz generadora. Si G es una matriz generadora G del $[n,k]$-código C sobre F. Entonces cada elemento $x \in C$ es una combinación de las filas de G, es decir que $x = u_1G_1 + \cdots + u_kG_k$, donde $u_1, \ldots u_k \in F$ y $G_1, \ldots, G_k$ son filas de G. En otras palabras, C es el espacio fila de la matriz G. Así, tenemos el siguiente resultado.

Teorema 8. Si C es un $[n,k]$-código sobre F y G es una matriz generadora C, entonces $C = \{uG |\ u \in F^k\}$.

Esta representación de C nos provee de un esquema de *codificación*. Si G es una matriz generadora de un $[n,k]$-código C sobre $F = \mathbb{F}_q$. La matriz G determina una aplicación biyectiva $\varepsilon : F^k \to C$ dada por $u \mapsto uG$. Usaremos esta aplicación para representar q^k distintos mensajes por palabras-código. Primero, adoptaremos algunos esquemas fijos por los cuales los q^k vectores en F^k son identificados con los q^k mensajes, y entonces usaremos la aplicación codificadora ε. Para nuestros propósitos, sin embargo, la manera en que los elementos de F^k son identificados con los actuales mensajes es inmaterial. Así pues, podemos considerar al mismo F^k como el conjunto de mensajes. Nos referiremos a los elementos de F^k como *palabras-mensaje*. De esta manera, cada k-tupla palabra-mensaje u es codificada como una n-tupla palabra-código uG. Nos referiremos al numero $n-k$ como la redundancia del código C, y a k/n como su tasa de transmisión.

Consideremos el sistema lineal homogéneo $GX = 0$, es decir, el sistema de k ecuaciones lineales en n incógnitas:

$$\begin{aligned} g_{11}x_1 + \cdots + g_{1n}x_n &= 0 \\ &\vdots \\ g_{k1}x_1 + \cdots + g_{kn}x_n &= 0 \end{aligned}$$

Sea N el conjunto de todas las soluciones. Como las filas de la matriz G son linealmente independientes, el rango de G es igual a k. Entonces, por el Teorema rango-nulidad del álgebra lineal, N es un subespacio de F^n de dimensión $n-k$. A N se le llama el *espacio nulo* de G.

La matriz G no es la única matriz generadora de C, puesto que una base de C puede ser elegida de muchas maneras. Pero mostraremos que el espacio nulo N es determinado únicamente por C y es independiente de la particular elección de la matriz generadora G. Sea $y = [y_1 \cdots y_n] \in N$. Entonces

$$G_i y^T = g_{i1}y_1 + \cdots + g_{in}x_n = 0$$

para cada $i = 1, \ldots, k$, de donde $Gy^T = 0$. Por lo tanto, para cada $x = uG \in C$, $u \in F^k$, tenemos $xy^T = uGy^T = 0$. Recíprocamente, sea $y \in F^n$ tal que $xy^T = 0$ para cada $x \in C$. Entonces en particular, $G_i y^T = 0$ para cada $i = 1, \ldots, k$, por lo que $y \in N$. Así $N = \{y \in F^n |\ xy^T = 0 \text{ para todo } x \in C\}$. Esto prueba que N es independiente de la matriz G que se elija para generar C. Como N es un subespacio de F^n de dimensión $n-k$, N es un $[n, n-k]$-código lineal sobre F. Llamaremos a N el *código dual* de C y lo denotaremos por $C^\perp$.

Para cualesquiera vectores $x, y \in F^n$, denote $x \cdot y$ su producto interno; esto es,

$$x \cdot y = x_1y_1 + \cdots x_ny_n = xt^T = yx^T = y \cdot x.$$

Entonces podemos definir el código dual como sigue:

DEFINICIÓN. Si C es un $[n,k]$-código sobre F, entonces el *código dual* de C es definido como

$$C^{\perp} = \{y \in F^n |\, x \cdot y = 0 \text{ para todo } x \in C\}.$$

Por ejemplo si $F = \mathbb{Z}_2$, entonces $010111 \cdot 010111 = 0$, y así nos encontramos con la sorprendente posibilidad de que el producto punto de un vector no nulo con sigo mismo a veces pueda ser cero, en contraste con lo que sucede $F = \mathbb{R}$ (numeros reales). Por lo tanto, el producto punto no se comporta como la longitud de un vector. Pero aun así es un concepto útil.

Dos vectores $x, y \in F^n$ se dirá que son ortogonales si $x \cdot y = 0$. De esta manera cada vector en $C^{\perp}$ es ortogonal a cada vector en C. Un código lineal C se llamara *auto-ortogonal* si cada vector en C es ortogonal a si mismo y a cualquier otro vector en C. En otras palabras, C es auto-ortogonal si $C \subset C^{\perp}$.

Ahora mostraremos que la relación entre C y $C^{\perp}$ es simétrica; esto es, es código dual de $C^{\perp}$ es C. Sea $x \in C$, entonces $x \cdot y = 0$ para cada $y \in C^{\perp}$, por lo tanto $x \in (C^{\perp})^{\perp}$, así $C \subset (C^{\perp})^{\perp}$. Vimos anteriormente que si C es un subespacio de dimensión k, entonces $C^{\perp}$ es un subespacio de dimensión $n - k$. En consecuencia $(C^{\perp})^{\perp}$ tiene dimensión $n - (n - k) = k$. Así las dimensiones de C y $(C^{\perp})^{\perp}$ son iguales a k. Esto prueba que $C = (C^{\perp})^{\perp}$. Registremos este resultado en el siguiente teorema.

Teorema 9. Si C es un $[n,k]$-código, entonces $C^{\perp}$ es un $[n, n-k]$-código y $(C^{\perp})^{\perp}$.

Ejemplo 8. Encontraremos el dual del código binario $C = \{0000, 1111\}$, y mostraremos que C es auto-ortogonal.

C es un $[4,1]$-código sobre $\mathbb{F}_2$. Revisando el producto interno de los vectores en $(\mathbb{F}_2)^4$ con cada vector en C, encontramos que

$$C^{\perp} = \{0000, 1100, 1010, 1001, 0110, 0101, 0011, 1111\}.$$

Verificamos que $C^{\perp}$ es un $[4,3]$-código. Claramente, $C \subset C^{\perp}$, por lo que C es auto-ortogonal.

Ahora definiremos la segunda matriz de las dos matrices asociadas con un código lineal.

DEFINICIÓN. Si C es un $[n,k]$-código lineal, y H es una matriz generadora del código dual $C^{\perp}$. Entonces nos referiremos a H como la *matriz verificadora de paridad* del código C. O abreviadamente *matriz paridad*.

De desprende de la definición que si G es una matriz generadora del código C, entonces G es la matriz paridad del código dual $C^{\perp}$. Puesto que el dual de un $[n,k]$-código es un $[n, n-k]$-código, podemos inferir que la matriz paridad de un $[n,k]$-código C es la matriz H de tamaño $(n-k) \times n$ cuyas filas forman una base para $C^{\perp}$. Vimos que si G

es una matriz generadora de C, entonces el espacio nulo de G es $C^{\perp}$. Ahora H es una matriz generadora de $C^{\perp}$, y por tanto el espacio nulo de H es $(C^{\perp})^{\perp} = C$. Entonces $x \in C$ si y sólo si $Hx^T = 0$ o, equivalentemente, $xH^T = 0$. Todo esto muestra que una matriz paridad determina completamente el código, y así hemos probado el siguiente resultado.

Teorema 10. Si C es un $[n, k]$-código sobre F y H es una matriz paridad de C, entonces

$$C = \{x \in F^n | \, xH^T = 0 = Hx^T\}.$$

La condición $Hx^T = 0$ da un sistema homogéneo de $n - k$ ecuaciones lineales para las componentes de x. Nos referiremos a ellas como las ecuaciones de paridad.

El siguiente teorema provee una relación mutua entre la matriz generadora y la matriz paridad de un código lineal.

Teorema 11. Si C es un $[n, k]$-código. Si G es la matriz generadora y H la matriz paridad de C, entonces

$$GH^T = 0 = HG^T.$$

Recíprocamente, supongamos que G es una matriz de tamaño $k \times n$ y de rango k, y H es una matriz de tamaño $(n-k) \times n$ y de rango $n-k$, tales que $GH^T = 0$. Entonces H es la matriz de paridad del código C, si y sólo si G es la matriz generadora de C.

Demostración. Por el teorema 10, $xH^T = 0$ para cada $x \in C$, entonces, en particular, $G_i H^T = 0$ para cada fila G_i de G, y $GH^T = 0$. Tomando la transpuesta, conseguimos $HG^T = 0$. Esto prueba la primera parte del teorema.

Para probar la segunda parte, sean G y H las matrices mencionadas, con $GH^T = 0$, y supongamos que H es la matriz de paridad de C. Entonces $G_i H^T = 0$ para cada $i = 1, \ldots, k$. Por lo tanto $G_1, \ldots, G_k \in C$. Puesto que el rango de G es k, $G_1, \ldots, G_k$ son linealmente independientes y en consecuencia forman una base de C. Esto prueba que G es una matriz generadora de C. La recíproca se prueba de manera similar. □

Ejemplo 9. Encontraremos la matriz generadora y la matriz de paridad del código binario $C = \{000, 111\}$.

La matriz $G = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$ es la unica matiz generadora de C, Para encontrar la matriz de paridad H, primero encontraremos el código dual $C^{\perp}$. Examinando el producto interno de todos los vectores en $(\mathbb{F})^3$ con los vectores en C, encontramos que $C^{\perp} = \{000, 110, 011, 101\}$. En este punto, cualquiera dos vectores en $C^{\perp}$ forman una base. Por lo que podemos tomar

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

Ejemplo 10. Encontraremos el código lineal binario C con matriz de paridad

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

además la matriz generadora G de C. y la matriz de paridad del código binario $C = \{000, 111\}$. También encontraremos el código dual $C^{\perp}$.

Como H es una matriz de tamaño 2×5, sabemos que C es un $[5, 3]$-código. Además, C es el espacio nulo de la matriz H. Por lo que debemos resolver el sistema lineal homogéneo

$$\begin{aligned} x_1 + x_2 + x_3 + _4 + x_5 &= 0 \\ x_1 + x_3 + x_4 &= 0 \end{aligned}$$

Tomaremos a x_1, x_2 y x_3 como parámetros. Entonces $x_4 = x_1 + x_3$ y $x_5 = x_2$. En consecuencia la solución general del sistema lineal es

$$\begin{aligned} x &= \begin{bmatrix} x_1 & x_2 & x_3 & x_1 + x_3 & x_2 \end{bmatrix} \\ &= x_1 \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \end{bmatrix} + x_3 \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} G \end{aligned}$$

Donde a x_1, x_2, x_3 se les asigna valores arbitrarios (0 o 1) y

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

De esta manera G e una matriz generadora para el código C. Asignando todos los posibles valores a x_1, x_2, x_3, obtenemos

$$C = \{00000, 10010, 01001, 00110, 11011, 10100, 01111, 11101\}.$$

Ahora, H es un generador del código dual $C^{\perp}$; por lo tanto

$$C^{\perp} = \{00000, 11111, 10110, 01001\}.$$

Anteriormente observamos que una matriz generadora del código lineal C no es única (salvo en el caso trivial de un código binario de dimensión 1). Sea G una matriz generadora del C, y sea G' la matriz obtenida como resultado de aplicar operaciones elementales sobre las filas de G. Entonces cada fila de G es una combinación lineal de las filas de G', y recíprocamente. Entonces G y G' tiene el mismo espacio de filas C. Por lo tanto G' también es una matriz generadora de C. Recíprocamente, si G y G' son matrices generadoras de C, entonces cada una de ellas puede obtenerse aplicando operaciones elementales sobre las filas de la otra. En particular, sea C es un $[n, k]$-código, y suponiendo que las primeras k columnas de G son linealmente independientes. Entonces, aplicando operaciones elementales de filas, podemos transformar G en una matriz escalonada de la forma $G^* = \left[I_k \vdots A\right]$, donde I_k es la matriz identidad de orden k, y A es alguna matriz de tamaño $k \times (n - k)$. Nos referiremos a G^* como la *matriz generadora canónica* de C. Ahora sea $H^* = \left[-A^T \vdots I_{n-k}\right]$. Entonces H^* es una matriz de tamaño $(n - k) \times n$, de rango $n - k$, y

$$G^*(H^*)^T = \left[I_k \vdots A\right] \begin{bmatrix} -A^T \\ \cdots \\ I_{n-k} \end{bmatrix} = -A + A = 0.$$

Por lo tanto, por el teorema 11, H^* es la matriz de paridad de C. Nos referiremos a H^* como la *matriz paridad canónica* de C. Así tenemos el siguiente resultado:

Teorema 12. Sea C es un $[n,k]$-código. Si C tiene matriz generadora canónica $G = \left[I_k \vdots A\right]$, entonces $H = \left[-A^T \vdots I_{n-k}\right]$ es la matriz paridad canónica de C. Recíprocamente, si $H = \left[B \vdots I_{n-k}\right]$ es una matriz paridad de C, entonces $G = \left[I_k \vdots -B^T\right]$ es una matriz generadora de C.

La matriz generadora que encontramos par el código del ejemplo 10 esta en la forma canónica, y podemos obtenerla más fácilmente usando el teorema 12.

Ejemplo 11. Usando el teorema 12 encontraremos la matriz generadora canónica del código C del ejemplo 10.

La matriz de paridad dada

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

es transformada (por intercambio de filas y la adición de la primera fila a la segunda) a la forma canónica

$$H^* = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix} = \left[B \vdots I_2\right].$$

Por lo tanto la matriz generadora canónica de C es

$$G^* = \left[I_3 \vdots -B^T\right] = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Debemos remarcar que las filas de G son la base para un subespacio k-dimensional del espacio de todos los vectores de longitud n. Este subespacio es nuestro código lineal C. En otras palabras, cada palabra-código se puede expresar de forma única como combinación lineal de las filas de G. Si usamos una matriz $G = bigl[I_k \vdots A\right]$ para construir un código, las primeras k columnas determinan las palabras-código (o símbolos de información), las restantes $n-k$ columnas proveen la redundancia (símbolos de verificación).

Ejemplo 12. El código en la segunda mitad del ejemplo 1 era

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

las palabras-código 101010 y 010101 aparecen como filas en la matriz y la palabra-código 111111 es la suma de estas dos filas. Este es un $[6,2]$-código.

Ejemplo 13. El código en el ejemplo 2 fue

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

Por ejemplo, la palabra-código 11001001 es la suma en modulo 2 de la primera, segunda y quinta filas, por tanto es obtenida por multiplicar $(1,1,0,0,1,0,0)$ a G. Este es un $[8,7]$-código.

Ejemplo 14. En el ejemplo 4, la matriz es

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

Y como se desprende de su nombre, este es un $[7,4]$-código.

Hasta ahora todo se a visto desde la perspectiva del $[n,k]$-código lineal C y su matriz generadora G, el siguiente resultado nos muestra cual es el panorama desde la perspectiva de $C^{\perp}$, además de mostrar porque se le llama código dual.

COROLARIO 13. Si C es un $[n,k]$-código lineal con matriz generadora $G = \left[I_k \vdots A\right]$, entonces $C^{\perp}$ es un $[n, n-k]$-código lineal con matriz generadora $H = \left[-A^T \vdots I_{n-k}\right]$. Además, G es una matriz paridad para $C^{\perp}$.

Demostración. Se sigue de la definición de $C^{\perp}$, el teorema 9, y el teorema 12. □

DEFINICIÓN. Un código C es llamado *auto-dual* si $C = C^{\perp}$.

El código de Golay $\mathcal{G}_{23}$ que se vera en el capitulo 6 es un importante ejemplo de código auto-dual.

Ejemplo 15. Sea C un código binario con matriz generadora

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

El corolario dice que $C^{\perp}$ tiene matriz generadora

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

Esta es G con las filas intercambiadas, así las filas de G y las filas de H generan el mismo subespacio. Por tanto, $C = C^{\perp}$, lo que nos dice que C es auto-dual.

Mencionamos anteriormente que una matriz generadora G es un $[n,k]$-código C determina una aplicación decodificadora $e : F^k \to F^n$ que a cada palabra-mensaje $u \in F^k$ le hace corresponder una palabra-código $x = uG \in C$. Esta codificación adopta una forma particularmente simple cuando G es canónica. Si $G = \left[I_k \vdots A\right]$, entonces $x = uG = \left[u \vdots uA\right]$. Así, las primeras componentes de x forman una palabra del mensaje en si. Esta simple codificación añade a la palabra-mensaje $n-k$ símbolos de verificación dados por uA.

Si las k columnas de una matriz generadora G de un $[n, k]$-código C no son linealmente independientes, entonces C no tiene una matriz generadora canónica. Sin embargo, como G tiene rango k, existen k columnas linealmente independientes en G. Si reordenamos las columnas de G, podemos obtener una matriz G' cuyas primeras k columnas son linealmente independientes, pero entonces G' no es una matriz generadora de C. Si C' es el código lineal generado por G', los códigos C y C' se dice que son equivalentes. La definición general de códigos equivalentes es la siguiente:

DEFINICIÓN. Si C y C' son $[n, k]$-códigos sobre F. Los códigos C y C' se dice que son *equivalentes* si existe una aplicación biyectiva $f : C \to C'$ dada por

$$f(x_1, \ldots, x_n) = (\alpha_1 x_{\sigma(1)}, \ldots, \alpha_n x_{\sigma(n)})$$

donde $\alpha_1, \ldots, \alpha_n \in F$ son escalares no nulos, y σ es una permutación del conjunto $[1, \ldots, n]$.

La definición nos conduce al siguiente resultado.

Teorema 14. Si C y C' son $[n, k]$-códigos sobre F con matrices generadoras G y G', respectivamente. Los códigos C y C' son equivalentes si y sólo si una de sus matrices puede ser obtenida a partir de la otra a través de la aplicación de operaciones de los siguientes tipos:

(a) Operaciones elementales de fila.

(b) Permutación de columnas.

(c) Multiplicación de cualquier columna por un escalar no nulo en F.

Una similar propiedad se cumple para matrices de paridad de dos códigos equivalentes.

Distancia mínima

Ahora consideraremos la mínima distancia de un código lineal. Recordemos que la mínima distancia de un código C es definido por $d(C) = \min\{d(x, y) |\, x, y \in C, x \neq y\}$, donde $d(x, y)$ es la distancia de Hamming entre x y y. Para un arbitrario código, posiblemente no lineal, calcular la mínima distancia puede requerir el calculo de $d(x, y)$ para cada par de palabras-código, labor que en muchos casos es poco practica; pero para códigos lineales este calculo es muy sencillo gracias al siguiente concepto.

DEFINICIÓN. El *peso* de un vector $x \in F^n$, que escribira $\omega(x)$ es definido como el número de componentes no nulas en x.

Se sigue inmediatamente de la definición que para cualesquiera vectores $c, y \in F$.

$$d(x, y) = \omega(x - y), \quad \omega(x) = d(x, 0),$$

donde 0 denota el vector $0 \dots 0$.

Teorema 15. Sea C un código lineal. Entonces la mínima distancia de $d(C)$ es igual al más pequeño peso todos los códigos-palabra no nulos en C: esto es

$$d(C) = \min\{\omega(x) |\, x \in C, x \neq 0\}.$$

Demostración. Sea $d(C) = d$. Entonces existen $c, c' \in C$ tales que $d(c, c') = d$. En consecuencia $c - c' \in C$ y $\omega(c - c') = d$. Si x es cualquier vector no nulo en C. Entonces $\omega(x) = \omega(x - 0) = d(x, 0) \geq d$. Esto prueba que d es el más pequeño peso de cualquier código no nulo en C. □

Ahora veremos que si queremos encontrar la mínima distancia de un código lineal con M palabras-código, tenemos que comparar los pesos de $M - 1$ palabras-código no nulas. Pero para encontrar la distancia de un código general con M palabras-código, tenemos que comparar las distancias de $M(M - 1)/2$ pares de distintas palabras-código.

Como el simple criterio indica, encontrar la mínima distancia de un código lineal con M muy grande puede ser difícil. Además, un código lineal es con frecuencia definido especificando su matriz de paridad, y sin que las palabras-código sean explícitamente conocidas. El siguiente teorema, que da la relación entre la distancia mínima y la matriz de paridad de un código lineal, es por lo tanto de fundamental importancia.

Teorema 16. Sea H la matriz de paridad de un $[n, k]$-código C sobre F. Entonces $d(C)$ es igual al mínimo numero de columnas linealmente independientes en H. Consecuentemente, $d(C) \leq n - k + 1$.

Demostración. Si $H^1, \dots, H^n$ denotan las columnas de H, y $x \in F^n$. Entonces, por el teorema 10, $x \in C$ si y sólo si

$$Hx^T = x_1 H^1 + \cdots + x_n H^n = 0.$$

Si $d(C) = d$, entonces $\omega(x) \geq d$ para todo $x \in C$, y existen $c \in C$ con $\omega(c) = d$. Sean $c_{i_1}, \dots, c_{i_d}$ los componentes no nulos de C. Entonces

$$Hc^T = c_1 H^1 + \cdots + c_n H^n = c_{i_1} H^{i_1} + \cdots + c_{i_d} H^{i_d} = 0.$$

Por lo tanto H tiene columnas linealmente independientes, a saber $H^{i_1}, \dots, H^{i_d}$. Afirmamos que cualquier lista de menos de d columnas de H es linealmente dependiente. Supongamos que las columnas $H^{i_1}, \dots, H^{i_c}$ son linealmente dependientes, donde $c < d$. entonces existen escalares $\alpha_1, \dots, \alpha_c$, no todos nulos, tales que

$$\alpha_1 H^{i_1} + \cdots + \alpha_c H^{i_c} = 0.$$

Sea $x \in F^n$ tal que los componentes $i_1, \ldots, i_c$ de x son iguales a $\alpha_1, \ldots, \alpha_c$, respectivamente, y todas las restantes componentes son ceros. Entonces $Hx^T = 0$; por lo que $x \in C$. Pero $\omega(x) < d$, una contradicción. Esto prueba que d es el mínimo numero de columnas linealmente independientes en H.

Ahora H es una matriz de tamaño $(n-k) \times n$ y rango $n-k$. Por lo tanto, cualesquiera $n-k+1$ columnas de H son linealmente dependientes. Por lo tanto $d \leq n-k+1$. □

Ejemplo 16. Se encontrara la mínima distancia del $[n, n-1]$-código binario C con matriz de paridad

$$H = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \end{bmatrix}.$$

Claramente, cualesquiera dos columnas de H son linealmente dependientes. Por esto el mínimo numero de columnas linealmente dependientes es 2. Por lo tanto la mínima distancia del código C es 2.

Este código es conocido como un código de paridad y puede detectar un simple error. Las componentes de cada palabra-código x satisfacen la ecuación de paridad $x_1 + \cdots + x_n = 0 \pmod 2$. Por lo que cada palabra-código contiene un numero par de 1's.

Ejemplo 17. Se encontrara la mínima distancia del $[10, 9]$-código C sobre $\mathbb{F}_{11}$ definido por la matriz de paridad

$$H = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}.$$

Como en el ejemplo anterior, la mínima distancia de C es 2. Las palabras-código satisfacen la ecuación de paridad

$$xH^T = \sum_{i=1}^{10} ix_i = 0$$

Además de detectar un simple error en general, este código puede detectar un doble error cuando dos dígitos tienen que ser transpuestos. Supongamos que la palabra-código $x = x_1 \cdots x_{10}$ es transmitida, y que el vector recibido $y = y_1 \cdots y_{10}$ difiere de x solo en los dígitos j-ésimo y el k-ésimo deben ser transpuestos; es decir $y_j = x_k$, $y_k = x_j$, y $y_i = x_i$ para los restantes índices.

$$\begin{aligned} yH^T &= xH^t + (y-x)H^t = (y-x)H^t \\ &= \sum_{i=1}^{10} i(y_i - x_i) = (k-j)(x_j - x_k) \neq 0 \end{aligned}$$

Por lo tanto y no es una palabra-código, y así el error es detectado.

Este código es de interés practico porque es el usado para el International Standard Book Numbers (ISBN) ya comentado en el ejemplo 5.

Ejemplo 18. Se encontrara la mínima distancia del $[10, 8]$-código C sobre $\mathbb{F}_{11}$ definido por la matriz de paridad

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}.$$

Esta claro que ninguna columna en H es múltiplo escalar de otra. Así que cualesquiera dos columnas son linealmente independientes. Por eso $d > 2$. Por otro lado, el teorema 15 nos dice que $d \leq n-k+1 = 3$. Esto implica que C código que corrige un solo error.

Decodificación

Ahora consideraremos el *proceso de decodificación* para un código lineal. Como ya explicamos, el principio general de decodificación es encontrar la palabra-código mas cercana al vector recibido. Con este fin en mente, preparamos una tabla que dará las más próximas palabras-código para cada posible vector recibido. La estructura algebraica de un código lineal como un subespacio proveerá un conveniente método para preparar tal tabla. Si C es un subespacio de F^n, entonces C es un subgrupo del grupo aditivo F^n. Recordemos que para cada $a \in F^n$, el conjunto

$$a + C = \{a + c | \, c \in C\}$$

es llamado una **clase** de C. Dos elementos $x, y \in F^n$ pertenecen a la misma clase si y sólo si $x - y \in C$. Estas clases forman una partición del conjunto F^n. Por lo tanto F^n es la unión disjunta de distintas clases de C.

Sea y cualquier vector en F^n, y supongamos que $x \in C$ es la palabra-código más cercana a y. Ahora y se encuentra en la clase $y + C = \{y + c | \, c \in C\}$. Para todo $c \in C$, $d(y, x) \leq d(y, c)$; esto es , $\omega(y - x) \leq \omega(y - c)$. Por esto $y - x$ es el vector de menor peso en la clase que contiene y. Escribiendo $e = y - x$, tenemos $x = y - e$. De esta manera hemos probado el siguiente resultado.

Teorema 17. Sea $C \subset F^n$ un código lineal. Dado un vector $y \in F^n$, la palabra-código x más cercano a y esta dado por $x = y - e$, donde e es el vector de menor peso en la clase que contiene a y.

Si la clase que contiene a y tiene más de un vector de menor peso, entonces existe más de una palabra-código más cercana a y.

DEFINICIÓN. Si C es un código lineal en F^n. El *líder de clase* de una clase dada de C es definido como el vector con menor peso en esa clase.

Si existe más de un vector con el menor peso en una clase, elegiremos cualquiera de ellos para ser líder de clase. Recordar que si C es un código corrector de t errores, entonces un vector no puede estar a una distancia $\leq t$ de más de una palabra-código. Por lo tanto, si e es un líder de clase con peso $\leq t$, entonces e es el único vector de menor peso en esa clase.

Sea C un $[n, k]$-código sobre $F = \mathbb{F}_q$. Como F^n tiene q^n elementos y cada clase de C tiene q^k elementos, se deduce que deben haber q^{n-k} clases diferentes en C. Si denotamos a los lideres de clase por $e_1, e_2, \ldots, e_N$, donde $N = q^{n-k}$. Además, si suponemos que los lideres de clase tienen que ser numerados en orden ascendente de pesos; esto es, $\omega(e_i) \leq \omega(e_{i+1})$ para todo i. Así $e_1 = 0$ es el líder de la clase $C = 0 + C$. Sea $C = \{c_1, c_2, \ldots, c_M\}$, donde $M = q^k$ y $c_1 = 0$. Podemos disponer los q^n vectores en F^n en un arreglo de tamaño $N \times M$, como se muestra abajo, donde la (i, j)-entrada es el vector $e_i + c_j$. Así la i-ésima fila contiene los elementos de la clase $e_i + C$, con el líder de clase e_i como la primera entrada. La primera fila (superior) es C mismo. Este arreglo es

llamado la *tabla estándar* para el código C.

$$\begin{array}{ccccccc} e_1 = 0 = c_1 & c_2 & \cdots & c_j & \cdots & c_M \\ e_2 & e_2 + c_2 & \cdots & e_2 + c_j & \cdots & e_2 + c_M \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_i & e_i + c_2 & \cdots & e_i + c_j & \cdots & e_i + c_M \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ e_N & e_N + c_2 & \cdots & e_N + c_j & \cdots & e_N + c_M \end{array}$$

La tabla estándar es usada para decodificar como sigue: Supongamos que un vector $y \in F^n$ es recibido. Encontramos su posición en la tabla. Si y es la (i, j)-entrada en la tabla, entonces $y = e_i + c_j$. Como e_j es el vector con el menor peso en la clase, se sigue por el teorema 17 que la palabra-código más cercana a y es $x = y - e_i = c_j$. Así un vector y es decodificado como la palabra-código que esta a la cabeza de la columna en la que y aparece.

Si la palabra-código es transmitida y el vector y es recibido, entonces $e = y - x$ es llamado el vector error. De esta manera el líder de clase es el vector error para cada vector y que aparece en esa clase.

Ejemplo 19. Se construirá la tabla estándar para el código lineal binario C con matriz generadora

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

y después se decodificara el vector recibido 0101.

El código C generado por G es un $[4, 2]$-código, con palabras código dadas por

$$C = \{0000, 1011, 0110, 1101\}$$

Se construirá la tabla estándar con los $2^4 = 16$ vectores binarios de longitud 4 como sigue. Primero, listemos los cuatro elementos de código en la primera fila, comenzando por 0000. Entonces, del total de los 12 restantes vectores, elegimos el de menor peso (aquí tenemos varias opciones). Adicionando este vector a los vectores de la primera fila, se obtiene la segunda fila. De los restantes 8 vectores, de nuevo se elige uno de menor peso y se lo adiciona a la primera fila para obtener la tercera fila. Finalmente, se escoge el vector con menor peso de entre los restantes cuatro vectores, se adiciona este a la primera fila, y se obtiene la cuarta fila. Así, obtenemos lo siguiente:

$$\begin{array}{cccc} 0000 & 1011 & 0110 & 1101 \\ 1000 & 0011 & 1110 & 0101 \\ 0100 & 1111 & 0010 & 1001 \\ 0001 & 1010 & 0111 & 1100 \end{array}$$

Para decodificar el vector 0101, primero observamos que se encuentra en la $(2, 4)$-entrada de la tabla. Como 1101 es el primer vector de la cuarta columna, tenemos que 0101 es decodificado como la palabra-código 1101.

En este pequeño ejemplo, se debe notar que el proceso no es exactamente el mismo que la decodificación por el vecino más cercano, puesto que 0010 es decodificado como 0110, cuando el equivalente más cercano es 0000. El problema es que la mínima distancia de el código es 2, así que en general la corrección del error no es posible. Sin embargo,

si usamos un código que pueda corregir hasta t errores, este procedimiento aplicado correctamente decodifica todos los vectores que están a una distancia de a lo más t desde la palabra-código.

Ejemplo 20. Construiremos la tabla estándar para el código binario con matriz generadora

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

y después decodificaremos el vector recibido 01111.

El código C generado por G es un $[5, 2]$-código, con palabras código dadas por

$$C = \{00000, 10101, 01011, 11110\}$$

Por lo tanto existen $2^3 = 8$ distintas clases de C, así la tabla estándar consistirá de 8 filas. La mínima distancia de C es 3, por lo que $t = 1$. Además los cinco vectores de peso 1 producen cinco distintas clases. Para conseguir las restantes dos filas del arreglo, escogemos en cada caso un vector de peso 2 que no haya aparecido en las anteriores filas. Notar que cada una de las últimas filas tiene 2 vectores de menor peso.

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

Para decodificar el vector 01111, encontramos su posición en la tabla. Notamos que 01111 aparece en la tercera columna. La primera entrada en esa columna es 01011. Por lo tanto 01111 es decodificado como la palabra-código 01011.

Una tabla estándar es útil para decodificar cuando la longitud del código n es pequeño. Para valores grandes de n, este no es un método muy conveniente. Ahora describiremos un procedimiento más eficiente para decodificar.

DEFINICIÓN. Sea C un $[n, k]$-código sobre F con matriz de paridad H. Para cualquier vector $y \in F$, el *síndrome* de y, denotado por $S(y)$, es definido por como el vector

$$S(y) = yH^T.$$

Notar que el síndrome es definido con respecto a una especifica matriz de paridad H. Una matriz de paridad diferente dará un síndrome diferente. Para un $[n, k]$-código, $S(y)$ es un vector de longitud $n - k$. Este puede escribirse como un vector fila yH^T (como se estableció en la definición) o como un vector columna Hy^T. En el último caso, $S(y) = Hy^T = y_1H^1 + \cdots + y_nH^n$, donde $H^1, \ldots, H^n$ son las columnas de H.

Por el teorema 10, $S(y) = 0$ si y sólo si $y \in C$. Sean $y, y' \in F^n$, entonces $(y) = S(y')$ se cumple si y sólo si $(y-y')H^T = 0$, es decir que $y-y' \in C$. Por lo tanto, dos vectores tienen el mismo síndrome si y sólo si ellos se encuentran en la misma clase de C. De esta

manera existe una correspondencia uno a uno entre las clases de C y los síndromes. Una tabla de dos columnas mostrando a los lideres de clase e_i y los correspondientes síndromes $S(e_i)$ es llamada la tabla síndrome. Para decodificar un vector recibido y, calculamos sus síndrome $S(y)$ y entonces revisamos la tabla par encontrar el líder de clase e por lo que $S(e) = S(y)$. Entonces y es decodificado como $x = y - e$. Este procedimiento es conocido como decodificación por síndrome.

Ejemplo 21. Se escribirá la tabla síndrome para el código del ejemplo 19, y entonces decodificar 0101.

La matriz generadora del código dado en el ejemplo 19 esta en la forma canónica. En consecuencia por el teorema 12, su matriz de paridad es

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

Calculando eH^T para cada líder de clase e, obtenemos la siguiente tabla:

Líder de clase	Síndrome
0000	00
1000	11
0100	10
0001	01

El síndrome del vector dado 0101 es

$$S(y) = yH^T = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

De la tabla de síndromes encontramos que el líder de clase con síndrome 11 es $e = 1000$. Por lo tanto, decodificamos y como la palabra código $x = y - e = 1101$.

Ejemplo 22. Escribiremos la tabla síndrome para el código del ejemplo 20, y entonces decodificar 11010.

La matriz generadora del código dado en el ejemplo 20 esta en la forma canónica. En consecuencia por el teorema 12, su matriz de paridad es

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

Calculando eH^T para cada líder de clase e, obtenemos la siguiente tabla:

Líder de clase	Síndrome
10000	101
01000	011
00100	100
00010	010
00001	001
11000	110
10010	111

El síndrome del vector dado $y = 11010$ es

$$yH^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

De la tabla de síndromes encontramos que el líder de clase con síndrome 100 es $e = 00100$. Por lo tanto, decodificamos y como la palabra código $x = y - e = 11110$.

Decodificar por medio del síndrome requiere significativamente menos pasos que buscar la palabra-código más cercana al vector recibido. Sin embargo, para códigos más grandes este procedimiento es aun ineficiente para ser practico. En general, el problema de encontrar el vecino más cercano en un código lineal general es muy duro: en efecto, este es conocido como un problema NP-completo. Pese a esto, para ciertos tipos de códigos especiales, una decodificación eficiente es posible. Trataremos algunos ejemplos más adelante.

Capítulo 5

Códigos de Hamming

Ahora describiremos una importante clase de códigos lineales conocidos como códigos de Hamming. Los códigos de Hamming son una importante clase de simples códigos correctores de errores que pueden fácilmente codificar y decodificar. Ellos originalmente fueron usados para el control de errores en las llamadas telefónicas de larga distancia.

Primero definiremos el caso especial de códigos de Hamming binarios y después consideraremos los códigos de Hamming q-arios.

DEFINICIÓN. Sea r un entero positivo mayor que 1, y sea H una matriz de tamaño $r \times (2^r - 1)$ cuyas columnas son los distintos vectores no nulos en $\mathbb{F}_2^r$. Entonces el código con H como matriz verificadora de paridad es llamado código de Hamming binario y es denotado por $\text{Ham}(r, 2)$.

Notar que la definición no especifica cual es el orden en el que las columnas de H están escritas, pero cualquier permutación de las columnas dará un código equivalente. Así, para un r dado existen $(2^r - 1)!$ códigos de Hamming binarios equivalentes.

Como H es una matriz de tamaño $r \times (2^r - 1)$, $\text{Ham}(r, 2)$ es un código de longitud $n = 2^r - 1$ y dimensión $k = n - r = 2^-1 - r$. Por lo tanto, $\text{Ham}(r, 2)$ es un $[2^r - 1, 2^-1 - r]$-código. El parámetro $r = n - k$ representa la redundancia del código.

Por ejemplo si $r = 2$, entonces $\text{Ham}(2, 2)$ es un $[3, 2]$-código con matriz verificadora de paridad

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

Si escribimos las columnas tal que H este en la forma canónica, entonces por el teorema 12 la matriz generadora del código es $G = [1 \quad 1 \quad 1]$

Si tomamos $r = 3$, vemos que $\text{Ham}(3, 2)$ es un $[7, 4]$-código con matriz verificadora de paridad

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Aquí H no esta en la forma canónica. Pero si movemos las columnas apropiadamente para que las ultimas columnas de la matriz formen una matriz identidad de 3×3,

siendo irrelevante el orden de las restantes columnas. El resultado es la matriz verificadora de paridad H en su forma canónica. (En nuestro ejemplo, movemos la cuarta, segunda y primera columnas)

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

Observamos que la primera columna en H es la suma de las quinta y sexta columnas. Por lo tanto estas columnas son linealmente dependientes. Además, ninguna columna es múltiplo escalar de otra, así 3 es el mínimo numero de columnas linealmente dependientes en H. Así, por el teorema 16 la mínima distancia de $\text{Ham}(3,2)$ es 3. La dimensión del código es 4, por lo que el numero de palabras código en $\text{Ham}(3,2)$ es $2^4 = 16$, así en la notación del capitulo 1, $\text{Ham}(3,2)$ es un $(7,16,3)$-código, que como mostramos es perfecto. Esto prueba que $\text{Ham}(3,2)$ es un código perfecto con distancia mínima 3.

El código binario de repetición $\{111,000\}$ es también (trivialmente) un código perfecto con distancia mínima 3. Así vemos que $\text{Ham}(2,2)$ y $\text{Ham}(3,2)$ son ambos códigos perfectos con distancia mínima 3. Luego mostraremos que estas propiedades se cumplen para todos los códigos de Hamming, pero primero daremos una definición general de lo que es un código de Hamming q-ario, $\text{Ham}(r,q)$.

Sea $F = \mathbb{F}_q$ (donde q es la potencia de algún primo p) y r es un entero positivo mayor que 1. Entonces existen $q^r - 1$ vectores no nulos en F^r. Dado un vector no nulo $v \in F^r$, tenemos un conjunto de $q-1$ múltiplos escalares no nulos de v. Por lo que existen $n = (q^r-1)/(q-1)$ conjuntos (disjuntos por pares) de este tipo que forman una partición de los vectores no nulos de F^r. Si tomamos arbitrariamente un elemento de cada uno de estos conjuntos, obtenemos un conjunto de n vectores no nulos tales que ningún vector es múltiplo escalar de otro. En particular, podemos tomar todos estos vectores no nulos en F^r tal que la primera entrada no nula sea 1.

DEFINICIÓN. Sea $F = \mathbb{F}_q$ y r un entero positivo mayor que 1, y sea $n = (q^r-1)/(q-1)$. Sea H una matriz de tamaño $r \times n$ cuyas columnas son los vectores no nulos en F^r, tal que ninguna columna es múltiplo escalar de otra. Entonces el $[n, n-r]$-código con H como matriz verificadora de paridad es llamado código de Hamming q-ario y es denotado por $\text{Ham}(r,q)$.

Como en el caso de los códigos de Hamming binarios, $\text{Ham}(r,q)$ es una notación general para una clase de códigos equivalentes. Por ejemplo, si $r = 2$ y $q = p$ (primo), entonces una matriz verificadora de paridad para $\text{Ham}(2,p)$ es

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & 2 & 3 & \cdots & p-1 \end{bmatrix}.$$

El siguiente teorema establece la más importante propiedad de los códigos de Hamming.

Teorema 18. Ham(r, q) es un código perfecto con distancia mínima 3.

Demostración. Si H es una matriz verificadora de paridad de Ham(r, q), entonces algunas tres columnas en H son múltiplos escalares de

$$\begin{bmatrix}1\\0\\0\\\vdots\\0\end{bmatrix}, \quad \begin{bmatrix}0\\1\\0\\\vdots\\0\end{bmatrix}, \quad \begin{bmatrix}0\\0\\1\\\vdots\\0\end{bmatrix}$$

y son por lo tanto linealmente dependientes. Así, 3 es el mínimo numero de columnas linealmente dependientes en H. Entonces por el teorema 16, tiene distancia mínima igual a 3, lo que implica que este código solo puede corregir un error.

Para mostrar que el código es perfecto, invocamos al teorema 6, el numero M de palabras código en Ham(r, q) es q^{n-r}, donde $n = (q^r - 1)/(q - 1)$. Además, $t = 1$, por lo que

$$\begin{aligned} M \sum_{m=0}^{t} \binom{m}{n} (q-1)^m &= q^{n-r}(1 + n(q-1)) \\ &= q^{n-r}(1 + q^r - 1) \\ &= q^n \end{aligned}$$

Esto prueba que el código es perfecto. □

Podemos fácilmente calcular la matriz generadora G a partir de la matriz de paridad H. Como los códigos de Hamming son simples códigos correctores de errores, el método del síndrome para decodificar puede ser simplificado. En particular, el vector error e tendrá un peso de a lo más 1 y por tanto tendrá ceros en todas las posiciones salvo por un 1 en la j-ésima posición.

El algoritmo de decodificación de Hamming, que corrige hasta el error en un bit, es el que sigue:

1. Calcular el síndrome $s = yH^T$ para el vector recibido y. Si $s = 0$, entonces no hay errores. Devolvemos el vector recibido y se acabo.
2. Por otro lado, determinar la posición j de la columna de H que es la transpuesta del síndrome.
3. Cambiar el j-ésimo bit en la palabra recibida, y el código resultante es la salida.

Como a lo más hay un bit error en el vector recibido, el resultado será la palabra-código que buscábamos.

Ejemplo 23. El $[15, 11]$-código binario de Hamming tiene la matriz de paridad

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

Asumamos que el vector recibido es

$$y = 000010000011001.$$

El síndrome $s = yH^T$ es calculado, y resulta ser $s = 1111$. Notar que s es la transpuesta de la onceava columna de H, así cambiaremos el onceavo bit de y para conseguir decodificar la palabra como

$$000010000001001.$$

Como los 11 primeros bits dan la información, el mensaje original fue

$$00001000000.$$

Capítulo 6

Códigos Cíclicos

Ahora discutiremos una clase especial de códigos lineales, conocidos como códigos cíclicos. Estos códigos tienen interesantes propiedades dentro de la teoría de anillos y una estructura algebraica más rica que los códigos lineales en general. Muchos importantes códigos pertenecen a esta categoría.

Una aplicación $\sigma : F^n \to F^n$ dada por

$$\sigma(a_1, a_2, \ldots, a_n) = (a_n, a_1, \ldots, a_{n-1})$$

es llamada una permutación cíclica. Es fácil ver que σ es una aplicación lineal; esto es, que para todo $a, b \in F^n$ y $\lambda \in F$,

$$\sigma(a + b) = \sigma(a) + \sigma(b) \text{ y } \sigma(\lambda a)\lambda\sigma(a).$$

DEFINICIÓN. Un código lineal $C \subset F^n$ es llamado un código cíclico si $\sigma(a) \in C$ para todo $a \in C$.

Por ejemplo, el código $C = \{000, 110, 011, 101\}$ es un código cíclico binario.

Teorema 19. Sea G la matriz generadora del $[n, k]$-código lineal C. Entonces C es un código cíclico si y sólo si $\sigma(G_i) \in C$ para cada fila G_i de G.

Demostración. Si G es cíclico, entonces $\sigma(a) \in C$ para todo $a \in C$. Por lo que, en particular, $\sigma(G_i) \in C$ para cada $i = 1, \ldots, k$. Recíprocamente, supongamos que $\sigma(G_i) \in C$ para cada fila G_i de G. Sea $a \in C$, entonces $a = \lambda_1 G_1 + \cdots + \lambda_k G_k$ par algunos escalares $\lambda_1, \ldots, \lambda_k$. Por lo que

$$\sigma(a) = \sigma(\lambda_1 G_1 + \cdots + \lambda_k G_k) = \lambda_1\sigma(G_1) + \cdots + \lambda_k\sigma(G_k) \in C.$$

Por lo tanto, C es cíclico. □

Ejemplo 24. Mostremos que el código C con la matriz generadora

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

es cíclico.

Claramente, $\sigma(G_1) = G_2$, $\sigma(G_2) = G_3$ y $\sigma(G_3) = G_4$. Además,

$$\sigma(G_4) = 1000110 = G_1 + G_2 + G_3.$$

Así $\sigma(G_) \in C$ para cada $i = 1, 2, 3, 4$. Por lo tanto C es cíclico.

De aquí en adelante numeraremos las componentes de un vector en F^n como $0, 1, \ldots, n-1$; esto es, si $a \in F^n$ escribiremos $a = (a_0, a_1, \ldots, a_{n-1})$. Entonces a cada vector $a \in F^n$ le corresponde el polinomio $a(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$. $F[x]_n$ denotara el conjunto de todos los polinomios en x de grado menor que n sobre el campo F; esto es

$$F[x]_n = \{a_0 + a_1x + \cdots + a_{n-1}x^{n-1} |\, a_0, a_1, \ldots, a_{n-1} \in F\}.$$

Claramente, $F[x]_n$ es un espacio vectorial de dimensión n sobre F. La aplicación $a \mapsto a(x)$ es un isomorfismo entre los espacios F^n y $F[x]_n$. Por esto, de aquí en adelante se identificara F^n con $F[x]_n$ y se tratara a cada vector $a \in F^n$ como un polinomio $a(x) \in F[x]_n$; es decir

$$(a_0, a_1, \ldots, a_{n-1}) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}.$$

Ahora impondremos a $F[x]_n$ la estructura de anillo como sigue; Como es usual, sea $F[x]$ denota el anillo de los polinomios en x sobre F. Sabemos que, dado cualquier polinomio $p(x) \in F[x]$, podemos construir el anillo cociente $F[x]/\langle p(x)\rangle$ donde $\langle p(x)\rangle$ denota el ideal principal generado por $p(x)$. Los elementos del anillo son las clases laterales de el ideal $\langle p(x)\rangle$. El anillo cociente es un campo si y sólo si $p(x)$ es irreducible sobre F. Además, si $p(x)$ es de grado n, entonces

$$\frac{F[x]}{\langle p(x)\rangle} = \{a_0 + a_1t + \cdots + a_{n-1}t^{n-1} |\, a_0, a_1, \ldots, a_{n-1} \in F\}$$

donde t denota la clase lateral $x + \langle p(x)\rangle$, con $p(t) = 0$.

Si tomamos $p(x) = x^n - 1$, entonces tendremos el anillo cociente

$$\frac{F[x]}{\langle x^n - 1\rangle} = \{a_0 + a_1t + \cdots + a_{n-1}t^{n-1} |\, a_0, a_1, \ldots, a_{n-1} \in F\}$$

donde t satisface la relación $x^n - 1 = 0$. Veras que este anillo no es un campo por que $p(x)$ se puede factorizar como $x^n - 1 = (x - 1)(x^{n-1} + \cdots + x + 1)$.

Si ahora hacemos un cambio en la notación y escribimos x en lugar de t, entonces el anillo $F[x]/\langle x^n - 1\rangle$ viene a ser $F[x]_n$. Así $F[x]_n$ se convierte en un anillo en el que la relación $x^n - 1 = 0$ se cumple, y como $F[x]_n$ es un espacio vectorial sobre F, $F[x]_n$ es un álgebra sobre F.

La multiplicación en el anillo $F[x]_n$ es modulo $\langle x^n - 1\rangle$. Para evitar cualquier confusión entre las multiplicaciones en los anillos $F[x]$ y $F[x]_n$ denotaremos a la última por $\odot$. Así, dados $a(x), b(x) \in F[x]_n$, escribiremos $a(x) \odot b(x)$ para denotar su producto en el anillo $F[x]_n$, y $a(x)b(x)$ denotara su producto en el anillo $F[x]$. Si $\deg a(x) + \deg b(x) < n$, entonces $a(x) \odot b(x) = a(x)b(x)$. Por otro lado $a(x) \odot b(x)$ es el residuo que queda de dividir $a(x)b(x)$ por $x^n - 1$. En otras palabras, $a(x) \odot b(x) = c(x)$, entonces $a(x)b(x) = c(x) + (x^n - 1)q(x)$ para algún polinomio $q(x)$. En la practica, para obtener $a(x) \odot b(x)$ simplificamos el resultado de multiplicar $a(x)b(x)$ colocando $x^n = 1$, $x^{n+1} = x$, etc.

En particular, considera el producto $x \odot a(x)$, en el anillo $F[x]$

$$xa(x) = x(a_0 + a_1x + \cdots + a_{n-1}x^{n-1}) = a_0x + a_1x^2 + \cdots + a_{n-1}x^n.$$

Sin embargo en el anillo $F[x]_n$

$$x \odot a(x) = a_{n-1} + a_0x + a_1x^2 + \cdots + a_{n-2}x^{n-1}.$$

Así vemos que la multiplicación por x en el anillo $F[x]_n$ corresponde a la permutación cíclica σ en F^n, es decir, que $x \odot a(x) = \sigma(a)(x)$.

Si $C \subset F^n$ es un código lineal. Como ya se vio, cada vector a en F^n se identifica son el polinomio $a(x)$ en $F[x]_n$, así $C \subset F[x]_n$. Ahora los elementos del código C se pueden tratar como palabras código o como códigos polinomiales. A partir del resultado ya probado anteriormente, conseguimos el siguiente teorema.

Teorema 20. Sea C un código lineal sobre F. Entonces C es cíclico si y sólo si $x \odot a(x) \in C$ para cada $a(x) \in C$.

En el siguiente teorema demostraremos una propiedad algebraica básica de los códigos cíclicos. Recordemos que un ideal en un anillo conmutativo R es un subconjunto no vacío A de R tal que para todo $a, b \in A$ y cualquier $r \in R$, se tiene que $a - b \in A$ y $ra \in A$.

Teorema 21. Un subconjunto C de $F[x]_n$ es un código cíclico si y sólo si C es un ideal del anillo $F[x]_n$.

Demostración. Supongamos que C es un código cíclico, entonces C es un código lineal sobre F, por lo que, para todo $a(x), b(x) \in C$ y cualquier $\lambda \in C$, se tiene que $a(x) - b(x) \in C$ y $\lambda a(x) \in C$, Además, como C es cíclico, $x \odot a(x) \in C$ para todo $a(x) \in C$. Por lo tanto, $x^2 \odot a(x) = x \odot (x \odot a(x)) \in C$, y así en adelante. Por lo tanto, para cada $r(x) = r_0 + r_1x + \cdots + r_{n-1}x^{n-1} \in F[x]_n$,

$$r(x) \odot a(x) = r_0 \odot a(x) + r_1x \odot a(x) + \cdots + r_{n-1}x^{n-1} \odot a(x) \in C.$$

Esto prueba que C es un ideal en el anillo $F[x]_n$.

Recíprocamente, supongamos que C es un ideal. Sean $a(x), b(x) \in C$ y $\lambda \in F$, entonces se tiene que $a(x) - b(x) \in C$ y como $\lambda \in F[x]_n$, resulta que $\lambda a(x) \in C$. Por esta razón C es un código lineal. Además, $r(x) \odot a(x) \in C$ para todo $r(x) \in F[x]_n$, y en particular $x \odot a(x) \in C$. Esto prueba que C es un código cíclico. □

Dado cualquier polinomio $p(x) \in F[x]_n$, entonces $\langle p(x) \rangle$ denota el ideal principal generado por $p(x)$ en el anillo $F[x]_n$; esto es

$$\langle p(x) \rangle = \{f(x) \odot p(x) |\ f(x) \in F[x]_n\}.$$

El siguiente teorema muestra que cada ideal en $F[x]_n$ es un ideal principal.

Teorema 22. Si C es cualquier ideal no nulo en el anillo $F[x]_n$, entonces

(a) Existe un único polinomio mónico $g(x)$ de mínimo grado en C.

(b) $g(x)$ divide a $x^n - 1$ en $F[x]$.

(c) Para todo $a(x) \in C$, $g(x)$ divide a $a(x)$ en $F[x]$.

(d) $C = \langle g(x) \rangle$.

Recíprocamente, supongamos que C es un ideal generado por $p(x) \in F[x]_n$, entonces $p(x)$ es un polinomio de mínimo grado en C si y sólo si $p(x)$ divide a $x^n - 1$ en $F[x]$.

Demostración. (a) Supongamos que $g(x)$ y $h(x)$ son polinomios mónicos de mínimo grado k en C. Sea $f(x) = g(x) - h(x)$, entonces $f(x) \in C$ y $\deg f(x) < k$. Si λ es el coeficiente principal en $f(x)$, entonces $\lambda^{-1} f(x)$ es un polinomio mónico de grado $< k$, una contradicción. Por lo tanto existe un único polinomio mónico $g(x)$ de mínimo grado en C.

(b) Por el algoritmo de la división en el anillo $F[x]$, existen polinomios $q(x)$ y $r(x)$ tales que

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

con $\deg r(x) < \deg g(x)$ o $r(x) = 0$. Por lo tanto, en $F[x]_n$ se tiene $q(x) \odot g(x) + r(x) = 0$, así $r(x) = -q(x) \odot g(x) \in C$. En consecuencia $g(x)$ es de grado mínimo en C, esto implica que $r(x) = 0$, por lo que $g(x)$ divide a $x^n - 1$ en $F[x]$.

(c), (d) Como C es un ideal en $F[x]_n$ y $g(x) \in C$, para cada $f(x) \in F[x]_n$, $f(x) \odot g(x) \in C$. Recíprocamente, sea $a(x) \in C$, entonces por el algoritmo de la división en $F[x]$,

$$a(x) = q(x)g(x) + r(x)$$

donde $\deg r(x) < \deg g(x)$ o $r(x) = 0$. Como $\deg a(x) < n$, $q(x)g(x) = q(x) \odot g(x)$. Por lo tanto $r(x) = a(x) - q(x) \odot g(x) \in C$. Por la minimalidad del grado de $g(x)$, $r(x) = 0$. Por lo que $a(x) = q(x)g(x)$. Así $g(x)$ divide a $a(x)$ en $F[x]$. Además, tenemos $C = \{f(x) \odot g(x) |\ f(x) \in F[x]_n\} = \langle g(x) \rangle$.

Para probar la última parte del teorema, supongamos que $p(x)$ divide a $x^n - 1$ y que $x^n - 1 = p(x)q(x)$. $a(x)$ es cualquier polinomio no nulo en $C = \langle p(x) \rangle$. Entonces $a(x) = f(x) \odot p(x)$ para algún $f(x) \in F[x]_n$. Por lo tanto

$$a(x) = f(x)p(x) + b(x)(x^{-}1) = \{f(x) + b(x)q(x)\}p(x).$$

para algún $b(x)$. Por lo tanto $\deg a(x) \geq \deg p(x)$, así $p(x)$ es de mínimo grado en C. Recíprocamente, si $p(x)$ es de mínimo grado en C, entonces por la parte (b) $p(x)$ divide a $x^n - 1$. □

Como consecuencia del anterior teorema vemos que los únicos ideales en el anillo $F[x]_n$ son los generados por los factores de $x^n - 1$. Así podemos obtener todos los códigos cíclicos de longitud n sobre F si encontramos todos los factores de $x^n - 1$. En el caso de factores triviales obtenemos códigos triviales. Cuando $g(x) = x^n - 1$, obtenemos $\langle g(x) \rangle = (0)$. Cuando $g(x) = 1$, tenemos $\langle g(x) \rangle = F[x]_n$.

Note que si $p(x)$ no divide a $x^n - 1$, entonces $p(x)$ no puede ser de mínimo grado en el ideal $\langle p(x) \rangle$. Por ejemplo, considerar $p(x) = x^{n-1} + 1 \in F[x]_n$. Entonces $x + 1 = x \odot (x^{n-1} + 1) \in \langle x^{n-1} + 1 \rangle$.

Ejemplo 25. Encontraremos todos los ideales no triviales de $F[x]_3$, donde $F = \mathbb{F}_2$. Además de obtener todos los códigos cíclicos binarios de longitud 3.

El polinomio $x^3 - 1$ puede ser factorizado como $x^3 - 1 = (x - 1)(x^2 + x + 1)$. Ahora bien $x - 1$ y $x^2 + x + 1$ son ambos irreducibles sobre $\mathbb{F}_2$, de esta manera los únicos factores no triviales de $x^3 - 1$ son $x - 1$ y $x^2 + x + 1$. Los ideales generados por ellos son

$$\langle x - 1 \rangle = \{0, 1 + x, x + x^2, 1 + x^2\}$$
$$\langle x^2 + x + 1 \rangle = \{0, 1 + x + x^2\}$$

Si escribimos los polinomios como vectores, obtenemos los siguientes códigos cíclicos $\{000, 110, 011, 101\}$ y $\{000, 111\}$.

DEFINICIÓN. Sea C un ideal no nulo en $F[x]_n$ y $g(x)$ el único polinomio mónico de mínimo grado en C. Entonces $g(x)$ es llamado el polinomio generador del código cíclico C.

Nota que si $C = \langle p(x) \rangle$ es el ideal generado por $p(x)$, entonces $p(x)$ es el polinomio generador de C si y sólo si $p(x)$ es mónico y divide a $x^n - 1$.

En el siguiente teorema, mostraremos que los códigos binarios de Hamming son (equivalentes a) códigos cíclicos. Recordemos que para cada $r > 1$, $\text{Ham}(r, 2)$ queda establecido para una clase de códigos equivalentes definidos por una matriz H verificadora de paridad de tamaño $r \times (2^r - 1)$ cuyas columnas son vectores no nulos en $\mathbb{F}_2^r$. Mostraremos que para un orden adecuado de las columnas de H obtenemos un código cíclico de Hamming $\text{Ham}(r, 2)$.

Teorema 23. Si $p(x)$ es un polinomio irreducible primitivo de grado r sobre $F = \mathbb{F}_2$. Sea $n = 2^r - 1$. Entonces el código cíclico con el polinomio generador $p(x)$ en el anillo $F[x]_n$ es $\text{Ham}(r, 2)$.

Demostración. Mostraremos que $p(x)$ divide a $x^n - 1$. El anillo cociente $K = F[x]/\langle p(x) \rangle$ es un campo de 2^r elementos, dado por

$$K = \{a_0 + a_1 t + \cdots + a_{r-1} t^{r-1} | \, a_0, a_1, \ldots, a_{n-1} \in F\}$$

donde t denota la clase lateral $x + \langle p(x) \rangle$ y satisface la relación $p(t) = 0$. El grupo multiplicativo K^* es de orden $2^r - 1 = n$, así $t^n = 1$. Por el algoritmo de la división,

$$x^n - 1 = q(x)p(x) + s(x)$$

donde $s(x) = 0$ o $\deg s(x) < r$. Colocando $x = t$, conseguimos $s(t) = 0$, que implica $s(x) = 0$. Esto prueba que $p(x)$ divide a $x^n - 1$.

Sea C el código cíclico en $F[x]_n$ con polinomio generador $p(x)$. Por el teorema 22, si $a(x)$ esta en C, entonces $a(x) = q(x)p(x)$ para algún $q(x)$, por lo que $a(t) = 0$. Recíprocamente, supongamos que $a(x) \in F[x]_n$ tal que $a(t) = 0$, entonces empleando el mismo

argumento que se uso para $x^n - 1$, podemos mostrar que $p(x)$ divide a $a(x)$, por lo que $a(x) \in C$. Así

$$C = \{a(x) \in F[x]_n |\, a(t) = 0\}.$$

Ahora escribiremos cada elemento $b_0 + b_1 t + \cdots + b_{r-1}t^{r-1} \in K$ como un vector columna $[b_o, b_1, \ldots, b_{r-1}]^T$, además sea

$$H = \begin{bmatrix} 1 & t & \cdots & t^{n-1} \end{bmatrix}$$

Como $p(x)$ es un polinomio primitivo, t es un elemento primitivo de K, así $1, t, \ldots, t^{n-1}$ son elementos no nulos y diferentes en K. Así H es una matriz de tamaño $r \times n$ cuyas columnas son todos los vectores no nulos y diferentes en $\mathbb{F}_2^r$.

Ahora, para cualquier $a(x) = a_0 + a_1 x + \cdots + a_{r-1}x^{r-1} \in F[x]_n$,

$$\begin{aligned} a(t) &= a_0 + a_1 x + \cdots + a_{r-1}x^{r-1} \\ &= H \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} \end{aligned}$$

Escribiendo el polinomio $a(x)$ como vector $a = (a_0, a_1, \ldots, a_{r-1})$, tenemos que

$$C = \{a \in F^n |\, Ha^T = 0\}.$$

Donde H es una matriz verificadora de paridad para el código C. Por definición, el código con matriz verificadora de paridad H es Ham$(r, 2)$. Esto prueba que C es un código Ham$(r, 2)$. □

Ejemplo 26. Se encontrara el polinomio generador del código cíclico

Consideremos el polinomio $p(x) = x^3 + x + 1$ sobre $\mathbb{F}_2$. Claramente, ni 0 ni 1 son raíces de $p(x)$, y por ello $p(x)$ es irreducible. Además el grupo multiplicativo del campo $\mathbb{F}_2^3$ es de orden 7 (que es primo). Por lo tanto cada elemento no nulo en $\mathbb{F}_2^3$ es primitivo. En consecuencia $p(x)$ es un polinomio primitivo. Por el teorema 23, el código cíclico con polinomio generador $x^3 + x + 1$ en el anillo $\mathbb{F}_2[x]_7$ es Ham$(3, 2)$.

De manera similar podemos mostrar que el código con polinomio generador $x^3 + x^2 + 1$ en el anillo $\mathbb{F}_2[x]_7$ es Ham$(3, 2)$.

El polinomio generador de un código cíclico C determina una matriz generadora así como una matriz verificadora de paridad para C. En el siguiente teorema obtendremos una matriz generadora.

Teorema 24. Si $C \subset F[x]_n$ es un código cíclico con polinomio generador

$$g(x) = g_0 + g_1 x + \cdots + g_r x^r$$

donde $g_r = 1$. Entonces C es de dimensión $n - r$. Además, la matriz de tamaño $(n - r) \times n$

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

es una matriz generadora de C.

Demostración. Por el teorema 22, $x^n-1 = g(x)h(x)$ para algún polinomio $h(x)$; por lo que $g_0 \neq 0$. Por lo tanto G es una matriz escalón por filas, y las filas de G son linealmente independientes. Escritos como polinomios, las filas de G son $g(x), xg(x), \ldots, x^{n-r-1}g(x)$. Sea $a(x) \in C$, por el teorema 22 $a(x) = q(x)g(x)$ para algún $q(x)$. Como $\deg a(x) < n$, tenemos que $\deg q(x) < n-r$, así $q(x)$ es de la forma $q(x) = q_0 + q_1x + \cdots + q_{n-r-1}x^{n-r-1}$, por lo que

$$a(x) = q_0g(x) + q_1xg(x) + \cdots + q_{n-r-1}x^{n-r-1}g(x).$$

De esta manera $a(x)$ puede ser expresado como una combinación lineal de las filas de G. Por lo tanto G es una matriz generadora de C. □

El polinomio generador de un código cíclico puede ser usado para decodificar. Se menciono en el anterior capitulo que si C es un $[n,k]$-código lineal con matriz generadora G, entonces una palabra mensaje $u \in F^k$ es decodificada como uG. Ahora supongamos que C es un código cíclico con polinomio generador $g(x)$, y sea G la matriz generadora obtenida por el teorema 24. Entonces $r = \deg g(x) = n-k$, y las filas de G son $g(x), xg(x), \ldots, x^{k-1}g(x)$.

Sea $u = (u_0, u_1, \ldots, u_{k-1})$, y $u(x) = u_0 + u_1x + \cdots + u_{k-1}x^{k-1}$. Entonces

$$uG = u_0g(x) + u_1xg(x) + \cdots + u_{k-1}x^{k-1}g(x) = u(x)g(x).$$

Así, el mensaje polinomial $u(x)$ es decodificado como $u(x)g(x)$.

DEFINICIÓN. Sea $g(x)$ un polinomio generador del código cíclico $C \subset F[x]_n$. Entonces el polinomio $h(x)$ tal que $x^n - 1 = g(x)h(x)$ es llamado el **polinomio verificador** de C.

Como $g(x)$ es mónico, $h(x)$ es también mónico. El siguiente teorema explica la nomenclatura.

Teorema 25. Si $C \subset F[x]_n$ es un código cíclico con polinomio verificador $h(x)$, sea $a(x) \in F[x]_n$. Entonces $a(x) \in C$ si y sólo si $a(x) \odot h(x) = 0$.

Demostración. Si $g(x)$ es el polinomio generador de C. Entonces $g(x)h(x) = x^n - 1$, por lo que $g(x) \odot h(x) = 0$. Sea $a(x) \in C$ entonces por el teorema 22, $a(x) = q(x)g(x)$ para algún $q(x)$, y por lo tanto $a(x) \odot h(x) = q(x)g(x) \odot h(x) = 0$. Recíprocamente, sea $a(x) \in F[x]_n$ tal que $a(x) \odot h(x) = 0$, entonces $a(x)h(x) = f(x)(x^n - 1)$ para algún $f(x)$, así $a(x)h(x) = f(x)g(x)h(x)$. Por lo tanto $a(x) = f(x)g(x)$, y de ahí que $a(x) \in C$. □

Si C es un $[n,k]$-código cíclico, entonces por el teorema 24 su polinomio generador $g(x)$ es de grado $n-k$, por lo que su polinomio verificador $h(x)$ es de grado k. Dado cualquier polinomio $f(x) = f_0 + f_1x + \cdots + f_mx^m$ de grado m, es polinomio recíproco de $f(x)$ es definido por

$$\overline{f}(x) = f_m + f_{m-1}x + \cdots + f_0x^m.$$

Los coeficientes en $\overline{f}(x)$ se toman de $f(x)$ en orden inverso. Formalmente, podemos escribir $\overline{f}(x) = x^m f(x^{-1})$. En particular, $\overline{g}(x) = x^{n-k}g(x^{-1})$ y $\overline{h}(x) = x^k h(x^{-1})$. Como $g(x)h(x) = x^n - 1$, se tiene que

$$\overline{g}(x)\overline{h}(x) = x^{n-k}g(x^{-1})x^k h(x^{-1}) = x^n(x^{-n} - 1) = 1 - x^n$$

por lo que $\overline{h}(x)$ divide a $x^n - 1$.

En el siguiente teorema, encontraremos una matriz verificadora de paridad para C y mostraremos que el código dual $C^\perp$ también es cíclico.

Teorema 26. Sea C el $[n, k]$-código cíclico con polinomio verificador

$$h(x) = h_0 + h_1 x + \cdot + h_k x^k$$

donde $h_k = 1$. Entonces

(a) La siguiente matriz

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \cdots & & \cdots & h_0 & 0 & \cdots & 0 \\ 0 & h_k & h_{k-1} & \cdots & & \cdots & h_1 & h_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & h_k & h_{k-1} & \cdots & & \cdots & h_0 \end{bmatrix}$$

es la matriz verificadora de paridad para C.

(b) El código dual $C^\perp$ es cíclico y es generado por el polinomio

Demostración. (a) Como H esta en la forma escalón por filas, sus filas son linealmente independientes. Afirmamos que cada fila de H es ortogonal a cada palabra código en C y por tanto un vector en $C^\perp$. Sea $a(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \in C$, entonces por el teorema 25, $a(x) \odot h(x) = 0$. Así el coeficiente de x^i a la izquierda del producto debe ser cero para cada $i = 0, 1, \ldots, n-1$. En particular, al igualar a cero el coeficiente de x^i para $i = k, k+1, \ldots, n-1$, obtenemos

$$a_{i-k} h_k + a_{i-k+1} h_{k-1} + \cdots + a_i h_0 = 0$$

para cada $i = k, k+1, \ldots, n-1$. Por lo tanto

$$H \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = 0$$

$a(x) = f(x)g(x)$, y de ahí que $a(x) \in C$. Así las filas de H son $n - k$ vectores linealmente independientes en el espacio dual $C^\perp$. Por lo tanto H es la matriz generadora de $C^\perp$ y en consecuencia la matriz verificadora de paridad de C.

(b) Como $\overline{h}(x)$ divide a $x^n - 1$, por el teorema 15, H es la matriz generadora del código cíclico $\langle \overline{h}(x) \rangle$, pero H es la matriz generadora de $C^\perp$, así $C^\perp = \langle \overline{h}(x) \rangle$. □

El siguiente ejemplo ilustra una aplicación de los teoremas 24 y 26.

Ejemplo 27. Escribiremos una matriz generadora y una matriz verificadora de paridad para el código cíclico de Hamming $\mathrm{Ham}(3, 2)$, obteniendo así el código completo.

Recordemos que $\text{Ham}(3,2)$ es un $[7,4]$-código lineal. Mostramos anteriormente que $g(x) = 1 + x + x^3$ es el polinomio generador del código C que es $\text{Ham}(3,2)$ cíclico. Por lo tanto, por el teorema 15 una matriz generadora de C es

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

Tomando todas las posibles combinaciones lineales de las filas de G, obtenemos

$$\begin{aligned} C = \{&0000000, 1101000, 0110100, 0011010, 0001101, 1011100, \\ &1110110, 1100101, 0101110, 0111001, 0010111, 1000110, \\ &0100011, 1111111, 1010001, 1001011\} \end{aligned}$$

El polinomio verificador de C es

$$h(x) = (x^7 - 1)/(x^3 + x + 1) = x^4 + x^2 + x + 1$$

Por lo tanto por el teorema 26, obtenemos

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

que es la matriz verificadora de paridad de C.

La matriz generadora y la matriz verificadora de paridad del código cíclico C dados por los teoremas 24 y 26 no están en la forma canónica. En general, para un código lineal, una matriz generadora G es transformada a la forma canónica aplicando operaciones elementales sobre las filas. Pero aquí, en el caso de un código cíclico, podemos obtener la forma canónica empleando el polinomio generador y el algoritmo de la división en $F[x]$.

Sea $g(x)$ el polinomio generador de un $[n,k]$-código cíclico C sobre F. Para cualquier polinomio $f(x) \in F[x]$, si $rem_{g(x)}(f(x))$ denota el residuo de dividir $f(x)$ por $g(x)$, por conveniencia escribiremos simplemente $r(f(x))$ en lugar de $rem_{g(x)}(f(x))$, así $f(x) = q(x)g(x) + r(f(x))$ para algún $q(x) \in F[x]$. Por lo tanto $f(x) - r(f(x)) = q(x)g(x) \in C$. En particular, considerar $r(x^j)$. Como $\deg g(x) = n - k$, tenemos $r(x^j) = x^j$ para $j < n - k$. Además, como $g(x)$ divide a $x^n - 1$, $r(x^{n+j}) = r(x^j)$ para todo $j \geq 0$. Así tenemos que calcular $r(x^j)$ sólo para $j = n - k, \ldots, n - 1$.

Definimos

$$G_i(x) = x^{i-1} + x^k \, r(x^{n-k+i-1})$$

para $i = 1, \ldots, k$. Claramente, $\deg G_i(x) < n$; por lo tanto $G_i(x) \in F[x]_n$. Además, $x^{n-k+i-1} - r(x^{n-k+i-1}) \in C$; por lo tanto $G_i(x) = x^k \odot (x^{n-k+i-1} - r(x^{n-k+i-1})) \in C$. Sea G la matriz de tamaño $k \times n$ cuya i-ésima fila es $G_i(x)$, escrita como vector fila, para $i = 1, \ldots, k$. Entonces

$$G = [I_k \vdots -A]$$

donde A es la matriz de tamaño $k \times (n-k)$ cuya i-ésima fila es $r(x^{n-k+i-1})$. Las filas de G son elementos de C y son linealmente independientes. Por lo tanto G es la matriz canónica generadora de C. Por el teorema 12 la matriz canónica de paridad de C es $H = [A^T \vdots I_{n-k}]$. De esta manera hemos probado el siguiente teorema.

Teorema 27. Sea $g(x)$ el polinomio generador de un $[n,k]$-código cíclico C sobre F. Sea A la matriz de tamaño $k \times (n-k)$ cuya i-ésima fila es $rem_{g(x)}(x^{n-k+i-1})$, para $i = 1, \ldots, k$. Entonces las matriz canónica generadora de C es $G = [I_k \vdots -A]$ y la matriz canónica de paridad de C es $H = [A^T \vdots I_{n-k}]$.

Ejemplo 28. Encontraremos las matrices canónicas generadora y verificadora de paridad del código cíclico Ham$(3,2)$.

Con $g(x) = 1 + x + x^3$, y para $k = 4$, $n = 7$, Dividiendo x^j por $g(x)$, par $j = 3,4,5,6$, obtenemos los residuos

$$r(x^) = 1 + x + x^2, \quad r(x^) = 1 + x + x^2, \quad r(x^) = 1 + x + x^2, \quad r(x^) = 1 + x + x^2.$$

De donde

$$A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

Por lo tanto

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Ahora consideremos la función síndrome para un código cíclico. Recordemos que si C es un $[n,k]$-código lineal sobre F con matriz verificadora de paridad H, entonces el síndrome de un vector $a \in F^n$ (con respecto a H) es $S(a) = aH^T$. El siguiente teorema proporciona el síndrome de un código cíclico con respecto a la matriz canónica verificadora de paridad.

Teorema 28. Sea C un $[n,k]$-código cíclico sobre F con polinomio generador $g(x)$. Sea H la matriz canónica verificadora de paridad de C. Entonces para cualquier $a \in F^n$,

$$S(a) = rem_{g(x)}(x^{n-k}a(x)).$$

Demostración. Por el teorema 27, $H = [A^T \vdots I_{n-k}]$, donde A es la matriz de tamaño $k \times (n-k)$ cuya i-ésima fila es $r(x^{n-k+i-1})$, para $i = 1, \ldots, k$. La i-ésima fila de matriz identidad I_{n-k} es x^{i-1}, para $i = 1, \ldots, n-k$. Por lo tanto, usando la relación $r(x^{n+j}) = r(x^j)$ vemos que la i-ésima fila de H^T es $r(x^{n-k+i-1})$, para $i = 1, \ldots, k$.

Sea $a = (a_0, a_1, \ldots, a_{n-1}) \in F$, así $a(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} \in F[x]_n$. Entonces

$$\begin{aligned} S(a) &= [a_0 \quad a_1, \ldots, a_{n-1}]H^T \\ &= \sum_{i=1}^{n} a_{i-1} r(x^{n-k+i-1}) \\ &= r\left(\sum_{i=1}^{n} a_{i-1} x^{n-k+i-1}\right) \\ &= r(x^{n-k}a(x)) \end{aligned}$$

□

Del teorema 28 vemos que para encontrar el síndrome de un código cíclico con respecto a la matriz canónica de paridad no necesitamos conocer esta matriz. Esto sugiere que podemos elegir una simple forma para el síndrome. Vimos en el capitulo 4 que par cualquier código lineal C, dos vectores tienen el mismo síndrome (con respecto a cualquier matriz de paridad) si y sólo si ellos se encuentran en la misma clase lateral de C. Si C es un código cíclico con polinomio generador $g(x)$, entonces los vectores a, b se encuentran en la misma clase lateral si y sólo si $g(x)$ divide $a(x) - b(x)$, es decir que $r(a(x)) = r(b(x))$. Por lo tanto, podemos definir el síndrome de un vector a como

$$S(a) = rem_{g(x)}(a(x)).$$

Este síndrome no es son respecto a la matriz de paridad de C obtenida arriba. Pero si tomamos H como la matriz cuya j-ésima columna es $r(x^{j-1})$, para $j = 1, \ldots, n$, entonces el síndrome con respecto a H es dado por $S(a) = r(a(x))$. Además, puede verificarse que la matriz H así definida es también una matriz de paridad.

De los varios resultados demostrados anteriormente, podemos observar que las matrices generadora G y de paridad H no juegan un rol directamente importante en los códigos cíclicos. El polinomio generador $g(x)$ únicamente determinado para el código cíclico C nos facilita el realizar todas las tareas para las que G y H son usadas en el caso de códigos lineales en general. Los elementos del código son de la forma $a(x)g(x)$. La decodificación es realizada por la aplicación $a(x) \mapsto a(x)g(x)$. El síndrome es dado por $rem_{g(x)}(a(x))$.

Tenemos todo listo para mostrar que los códigos binarios de Hamming son cíclicos. Ahora describiremos otros dos importantes códigos cíclicos, conocidos como *códigos de Golay*.

Ejemplo 29. *Código binario de Golay* G_{23}. El polinomio $x^{23} - 1$ puede ser factorizado sobre $\mathbb{F}_2$ como

$$x^{23} - 1 = (x - 1)g(x)g_1(x)$$

donde

$$\begin{aligned} g(x) &= x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1, \\ g_1(x) &= x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1. \end{aligned}$$

Estos dos polinomios son recíprocos uno del otro por lo que generan códigos equivalentes. Por el teorema 24, estos códigos son de dimensión 12. El $[23, 12]$-código cíclico

con polinomio generador $g(x)$ en $\mathbb{F}_2[x]_{23}$, o cualquier código equivalente a este, es llamado *Código binario de Golay* y es denotado por G_{23}. Ahora $g(x)$ es un elemento del código, y el número de términos no nulos en $g(x)$ es 7. Así, por el teorema 15, la mínima distancia del código es a lo más 7.

Ejemplo 30. *Código ternario de Golay* G_{11}. El polinomio $x^{11}-1$ puede ser factorizado sobre $\mathbb{F}_3$ como

$$x^{11}-1=(x-1)g(x)g_1(x)$$

donde

$$g(x)=x^5+x^4-x^3+x^2-1,$$
$$g_1(x)=x^5-x^3+x^2-x+1.$$

Estos dos polinomios generan códigos equivalentes de dimensión 6. El $[11,6]$-código cíclico con polinomio generador $g(x)$ en $\mathbb{F}_3[x]_{11}$, o cualquier código equivalente a este, es llamado *Código ternario de Golay* y es denotado por G_{11}. Como $g(x)$ tiene 5 términos no nulos, la mínima distancia del código es a lo más 5.

Ahora establecemos el siguiente teorema sin demostración.

Teorema 29. El $[23,12]$-código binario de Golay tiene distancia mínima 7. El $[11,6]$-código ternario de Golay tiene distancia mínima 5.

Con este resultado, ahora podemos fácilmente demostrar que ambos códigos de Golay son perfectos.

Teorema 30. El $[23,12,7]$-código binario de Golay y el $[11,6,5]$-código ternario de Golay, son ambos perfectos.

Demostración. Por el teorema 6 un (n,M,d)-código q-ario con $d=2t+1$ es perfecto si y sólo si

$$M\sum_{m=0}^{t}\binom{n}{m}(q-1)^m=q^n.$$

Mostraremos que esta condición se cumple para cada código de Golay.

Para el código binario de Golay, se tiene $q=2$, $n=23$, $t=3$, $M=2^{12}$, con lo que es fácil verificar que

$$2^{12}\left\{1+23+\binom{23}{2}+\binom{23}{3}\right\}=2^{23}$$

Para el código ternario de Golay, se tiene $q=3$, $n=11$, $t=2$, $M=3^6$, con lo que es fácil verificar que

$$3^6\left\{1+11\cdot 2+\binom{11}{2}2^2\right\}=3^{11}$$

Esto prueba que ambos códigos de Golay son perfectos. □

En el capitulo 4 mostramos que todos los códigos de Hamming son perfectos. Esencialmente los códigos de Hamming y los dos códigos de Golay son los únicos códigos perfectos. Y lo estableceremos en el siguiente teorema sin demostración.

Teorema 31. Cada código perfecto (no-trivial) corrector de errores simples tiene los parámetros de un código de Hamming.
Cada código perfecto (no-trivial) corrector de errores múltiples es equivalente, bien al $[23, 12, 7]$-código binario de Golay o bien al $[11, 6, 5]$-código ternario de Golay.

Capítulo 7

Códigos BCH

En este capitulo describiremos una clase especial de código cíclico, conocido como código BCH (llamado así por Bose, Chauduri y Hocquenghem). Los códigos BCH son una generalización de los métodos que usamos en el anterior capitulo para construir los códigos de Hamming. Mostraremos que si t es un elemento primitivo en el campo $\mathbb{F}_2^r$ y $p(x) \in \mathbb{F}_2^r[x]$ es un polinomio irreducible tal que $p(t) = 0$, entonces el código cíclico con polinomio generador $p(x)$ en $\mathbb{F}_2[x]_n$, donde $n = 2^r - 1$, es Ham$(r, 2)$. Un código BCH es construido a través de una generalización de esta técnica.

Antes de definir un código BCH, daremos un sumario de algunas propiedades importantes de los campos finitos y los polinomios irreducibles. Para cada primo p y entero positivo r, existe un único campo (salvo isomorfismo) de orden p^r, denotado por $\mathbb{F}_{p^r}$ o $GF(p^r)$. Si $r = 1$, entonces $\mathbb{F}_p = \mathbb{Z}_p$. Sea $F = \mathbb{F}_q$, donde $q = p^r$. Entonces el conjunto F^* de los elementos no nulos en F es un grupo cíclico de orden $q-1$ bajo la multiplicación. Por lo tanto $a^{q-1} = 1$; es decir que $a^q = a$ para cada $a \in F$. Para cualquier divisor n de $q - 1$, existe un elemento a de orden $o(a) = n$ en el grupo F^*. Tal elemento es llamado la n-ésima raíz primitiva de la unidad en el campo F. Si $o(a) = q - 1$, entonces a es generador del grupo cíclico F^* y es llamado elemento primitivo en F. La característica del campo F es p. Por lo que, par todo $a \in F$, $pa = 0$ y por consiguiente también $qa = 0$.

Dado un campo finito F y cualquier entero positivo m, existe un polinomio irreducible $p(x) \in F[x]$ de grado m. El campo $\mathbb{F}_{p^r}$ es obtenido construyendo el anillo cociente $\mathbb{Z}_p/\langle p(x)\rangle$, donde $p(x)$ es un polinomio irreducible de grado r en $\mathbb{Z}_p[x]$. Para mayor generalidad, empecemos con $F = \mathbb{F}_q$ y construyamos el campo $\mathbb{F}_{q^m}$ como el anillo cociente $\mathbb{F}_q[x]/\langle p(x)\rangle$, donde $p(x)$ es un polinomio irreducible de grado m en $\mathbb{F}_q[x]$. Si denotamos la clase lateral $x + \langle p(x)\rangle$ por t, entonces $p(t) = 0$ y

$$\mathbb{F}_{q^m} = \{a_0 + a_1 t + \cdots + a_{m-1}t^{m-1} |\, a_0, a_1, \ldots, a_{m-1} \in \mathbb{F}_q\}.$$

El campo $\mathbb{F}_{q^m}$ es llamado una extensión de grado m de $\mathbb{F}_q$.

Empleando el hecho de que $a^q = a$ para todo $a \in \mathbb{F}_q$ y $q\beta = 0$ para todo $\beta \in \mathbb{F}_{q^m}$ inmediatamente conseguimos el siguiente resultado:

Teorema 32. Sean $a_1, \ldots, a_n \in \mathbb{F}_q$ y $\beta_1, \ldots, \beta_n \in \mathbb{F}_{q^m}$, entonces

$$(a_1\beta_1 + \cdots + a_n\beta_n)^q = a_1\beta_1^q + \cdots + a_n\beta_n^q$$

Si $\alpha \in \mathbb{F}_{q^m}$, entonces existe un polinomio mónico $q(x) \in \mathbb{F}_q[x]$ de menor grado tal que $g(\alpha) = 0$. El polinomio $g(x)$ es irreducible sobre $\mathbb{F}_q$ y es llamado el polinomio minimal de de α sobre $\mathbb{F}_q$. Si $f(x) \in \mathbb{F}_q[x]$ es cualquier polinomio tal que $f(\alpha) = 0$, entonces, empleando el algoritmo de la división en $\mathbb{F}_q[x]$, podemos mostrar que $g(x)$ divide a $f(x)$. Además, $\deg g(x)$ divide a m. Si α es un elemento primitivo en $\mathbb{F}_{q^m}$, entonces $\deg g(x) = m$.

Por el teorema 32, se sigue que si $\alpha \in \mathbb{F}_{q^m}$ es una raíz del polinomio $f(x) \in \mathbb{F}_q[x]$, entonces α^q es también una raíz de $f(x)$. En particular, tenemos el siguiente resultado.

Teorema 33. Si $\alpha \in \mathbb{F}_{q^m}$, entonces $\alpha, \alpha^q, \alpha^{q^2}, \ldots$ tienen el mismo polinomio minimal sobre $\mathbb{F}_q$.

Un código BCH es definido como sigue: Sean c, d, q, n enteros positivos tales que $2 \leq d \leq n$, q es una potencia prima, y n es primo relativo a q. Si m es el mínimo entero positivo tal que $q^m \equiv 1 \bmod n$. Así n divide a $q^m - 1$. Sea ζ es una n-ésima raíz primitiva de la unidad en $\mathbb{F}_{q^m}$, y $m_i(x) \in \mathbb{F}_q[x]$ denota el polinomio minimal de ζ^i. Sea $g(x)$ el producto de distintos polinomios entre los $m_i(x)$, $i = c, c+1, \ldots, c+d-2$; esto es

$$g(x) = lcm\{m_i(x) |\, i = c, c+1, \ldots, c+d-2\}.$$

Como $m_i(x)$ divide a $x^n - 1$ para cada i, se sigue que $g(x)$ divide a $x^n - 1$. Sea C un código cíclico con polinomio generador $g(x)$ en el anillo $\mathbb{F}_q[x]_n$. Entonces C es llamado un *código BCH* de longitud n sobre $\mathbb{F}_q$ con distancia designada d.

Si $n = q^m - 1$ la previa definición, entonces el código BCH C es llamado *primitivo*. Si $c = 1$, entonces C es llamado un código BCH en el sentido estrecho.

Teorema 34. Si C es un código BCH de longitud n sobre $\mathbb{F}_q$ con distancia designada d. Entonces con la notación usada anteriormente,

$$C = \{v(x) \in \mathbb{F}_q[x]_n |\, v(\zeta^i) = 0 \text{ para todo } i = c, c+1, \ldots, c+d-2\}.$$

Equivalentemente, C es el espacio nulo de la matriz

$$H = \begin{bmatrix} 1 & \zeta^c & \zeta^{2c} & \cdots & \zeta^{(n-1)c} \\ 1 & \zeta^{c+1} & \zeta^{2(c+1)} & \cdots & \zeta^{(n-1)(c+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{c+d-2} & \zeta^{2(c+d-2)} & \cdots & \zeta^{(n-1)(c+d-2)} \end{bmatrix}.$$

Demostración. Sea $v(x) \in C$, entonces por el teorema 24, $v(x) = q(x)g(x)$ para algún $q(x)$, donde $g(x)$ es el polinomio generador de C. Por lo tanto $v(\zeta^i) = 0$ para todo $i = c, c+1, \ldots, c+d-2$. Recíprocamente, sea $v(x) \in \mathbb{F}_q[x]_n$ tal que $v(\zeta^i) = 0$ para todo $i = c, c+1, \ldots, c+d-2$. Entonces $m_i(x)$ divide a $v(x)$ para todo $i = c, c+1, \ldots, c+d-2$. Por lo tanto $g(x)$ divide a $v(x)$, por lo que $v(x) \in C$. Esto prueba la primera parte del teorema.

Para probar la segunda parte, sea $v(x) = v_0 + v_1 x + \cdots + v_{n-1}x^{n-1} \in \mathbb{F}_q[x]_n$. Entonces $v(\zeta^i) = 0$ se mantiene para todo $i = c, c+1, \ldots, c+d-2$ si y sólo si $Hv^T = 0$, donde $v = (v_0, v_1, \ldots, v_{n-1}) \in \mathbb{F}_q^n$. Esto prueba que C es el espacio nulo de H. □

Notamos que H es una matriz de tamaño $(d-1) \times n$ sobre $\mathbb{F}_{q^m}$. Cada elemento en $\mathbb{F}_{q^m}$ es de la forma $a_0 + a_1 t + \cdots + a_{m-1} t^{m-1}$, donde $a_0, a_1, \ldots, a_{m-1} \in \mathbb{F}_q$, por lo que pueden escribirse como vectores columna de longitud m. Por lo tanto H puede escribirse como una matriz de tamaño $m(d-1) \times n$ sobre $\mathbb{F}_q$. Las filas de H no son necesariamente linealmente independientes, por lo tanto H no es una matriz verificadora de paridad de C en el estricto sentido del término. A H la llamaremos matriz cuasi-verificadora de paridad. Como $rank(H) \leq m(d-1)$, se sigue que la $\dim C \geq n - m(d-1)$.

En el próximo teorema mostraremos que la distancia mínima de un código BCH es menor que o igual a su distancia designada. Mostramos en el teorema 16 que la distancia mínima $d(C)$ de un código lineal Ces igual al número mínimo de columnas linealmente independientes en la matriz verificadora de paridad H de C. La demostración de este teorema nos muestra que esta propiedad no depende de si las filas de H son linealmente independientes. Por lo tanto el teorema también se cumple cuando H es una matriz cuasi-verificadora de paridad.

Teorema 35. Si C un código BCH de distancia designada d, entonces $d(C) \geq d$.

Demostración. Si H es la matriz cuasi-verificadora de paridad de C dado en el teorema 34. Mostraremos que cualesquiera $d-1$ columnas de H son linealmente independientes. Sea K la matriz de tamaño $(d-1)\times(d-1)$ formada por las columnas con primeras entradas $\zeta^{i_1 c}, \zeta^{i_2 c}, \ldots, \zeta^{i_{d-1} c}$, respectivamente. Entonces

$$\det K = \zeta^{(i_1+i_2+\cdots+i_{d-1})c} \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \zeta^{i_1} & \zeta^{i_2} & \cdots & \zeta^{i_{d-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta^{i_1(d-2)} & \zeta^{i_2(d-2)} & \cdots & \zeta^{i_{d-1}(d-2)} \end{vmatrix}$$

El determinante del lado derecho es un determinante de Vandermonde. Ahora ζ es un elemento de orden n en el grupo multiplicativo de $\mathbb{F}_{q^m}$; por lo tanto

$$\zeta^{i_1}, \zeta^{i_2}, \ldots, \zeta^{i_{d-1}}$$

son todos distintos y en consecuencia $\det K \neq 0$. Por lo tanto las columnas de K son linealmente independientes. De esto se sigue que el numero mínimo de columnas linealmente independientes en H es mayor que $d-1$. Por lo tanto $d(C) \geq d$. □

Ahora presentaremos algunos ejemplos de códigos BCH. El más simple ejemplo es un código Reed-Solomon, que es definido como sigue: sea q cualquier potencia prima, y sea $n = q-1$, entonces $m = 1$. Sea ζ un elemento primitivo en $\mathbb{F}_q$, entonces el polinomio minimal de ζ sobre $\mathbb{F}_q$ es $x - \zeta$. Tomando $c = 1$ y d como cualquier entero positivo, $2 \leq d \leq n$. Entonces el código cíclico C con polinomio generador

$$g(x) = (x-\zeta)(x-\zeta^2)\cdots(x-\zeta^{d-1})$$

es un código primitivo BCH en el sentido estrecho de distancia designada d. Este es llamado un código Reed-Solomon. Como $\deg g(x) = d-1$, el código C tiene dimensión $k = n - d + 1$. Por el anterior teorema $d(C) \geq d$. Por otro lado el teorema 35 dice que $d(C) \leq n - k + 1 = d$, así $d(C) = d$. Por lo tanto C es un $[q-1, q-d, d]$-código.

A continuación mostraremos que los códigos binarios de Hamming y Golay son códigos BCH.

Para obtener Ham$(r, 2)$ como un código BCH, tomaremos $q = 2$ y $n = 2^r - 1$. Entonces $m = r$, así $\mathbb{F}_{q^m} = \mathbb{F}_{2^r}$. Sea ζ una n-ésima raíz primitiva de la unidad en $\mathbb{F}_{2^r}$. Entonces ζ es en efecto un elemento primitivo en $\mathbb{F}_{2^r}$. Sea $g(x) \in \mathbb{F}_2[x]$ el polinomio minimal de ζ. Entonces $g(x)$ es un polinomio primitivo de grado r. Ahora ζ y ζ^2 tienen el mismo polinomio minimal, por lo que $m_1(x) = m_2(x) = g(x)$. Así

$$g(x) = lcm\{m_i(x) |\, i = 1, 2\}.$$

Por lo tanto, por la definición, el código cíclico C con polinomio generador $g(x)$ es un código primitivo BCH en el sentido estricto con distancia designada 3. Por otro lado, por el teorema 23, C es Ham$(r, 2)$. Esto prueba que Ham$(r, 2)$ es un código BCH. Mostramos en el teorema 18 que la mínima distancia de Ham$(r, 2)$ es 3. Así en este caso $d(C) \geq d$.

Para el código binario de Golay, tomamos $q = 2$ y $n = 23$. Entonces $m = 11$, así $\mathbb{F}_{q^m} = \mathbb{F}_{2^{11}}$. Sea ζ una 23-ésima raíz primitiva de la unidad en $\mathbb{F}_{2^{11}}$ y sea $g(x)$ el polinomio minimal de ζ. Ahora $\zeta, \zeta^2, \zeta^{2^2}, \zeta^{2^3}, \ldots$ todos tienen el mismo polinomio minimal. Usando la relación $\zeta^{23} = 1$, tenemos $\zeta^{2^8} = \zeta^{256} = \zeta^3$. De esta manera $\zeta, \zeta^2, \zeta^3, \zeta^4$ tienen el mismo polinomio minimal $g(x)$. Así

$$g(x) = lcm\{m_i(x) |\, i = 1, 2, 3, 4\}.$$

El código cíclico C con polinomio generador $g(x)$ es un código BCH en el sentido estricto con distancia designada 5 sobre $\mathbb{F}_2$. Ahora $g(x)$ divide a $x^{23} - 1$ y el grado de $g(x)$ divide a $m = 11$, así $\deg g(x) = 11$. Por lo tanto C es el $[23, 12, 7]$-código binario de Golay descrito en el capitulo 6, en este caso $d(C) = 7 > d$.

Para el código ternario de Golay es también un código BCH. Sean $q = 3$ y $n = 11$. Entonces $m = 5$, así $\mathbb{F}_{q^m} = \mathbb{F}_{3^5}$. Sea ζ una 11-ésima raíz primitiva de la unidad en $\mathbb{F}_{3^5}$ y sea $g(x)$ el polinomio minimal de ζ. Entonces $\zeta, \zeta^3, \zeta^{3^2}, \zeta^{3^3}, \ldots$ todos tienen el mismo polinomio minimal. Como $\zeta^{11} = 1$, tenemos $\zeta^{27} = \zeta^5$ y $\zeta^{81} = \zeta^4$. De esta manera $\zeta^3, \zeta^4, \zeta^5$ tienen el mismo polinomio minimal $g(x)$. Así

$$g(x) = lcm\{m_i(x) |\, i = 3, 4, 5\}.$$

El código cíclico C con polinomio generador $g(x)$ es un código BCH de distancia designada 4. Ahora $g(x)$ es un polinomio irreducible de grado 5 que divide a $x^{11} - 1$. Por lo tanto C es el $[11, 6, 5]$-código ternario de Golay, y aquí también se tiene que $d(C) > d$.

El siguiente ejemplo ilustra como construir un código BCH de una longitud y distancia designada dados. El primer paso es tener un elemento primitivo ζ en el campo $\mathbb{F}_{q^m}$ para el que necesitamos un polinomio primitivo sobre $\mathbb{F}_q$ de grado m. Para referencia aquí tenemos algunos polinomios primitivos sobre $\mathbb{F}_2$ de grados 4 hasta 8:

$$x^4 + x + 1$$
$$x^5 + x^2 + 1$$
$$x^6 + x + 1$$
$$x^7 + x + 1$$
$$x^8 + x^4 + x^3 + x^2 + 1$$

Ejemplo 31. Supongamos que C es un código BCH en el sentido estricto de longitud 31 y distancia designada 5 sobre $\mathbb{F}_2$. Encontraremos su dimensión.

Aquí $q = 2$, $n = 15$, por lo que $m = 4$ y $2^4 - 1 = 15$. El polinomio

$$p(x) = x^4 + x + 1$$

es un polinomio primitivo irreducible sobre $\mathbb{F}_2$. Entonces podemos representar el campo $\mathbb{F}_{2^4}$ como

$$\mathbb{F}_{2^4} = \{a_0 + a_1\zeta + a_2\zeta^2 + a_3\zeta^3 |\, a_0, a_1, a_2, a_3 \in \mathbb{F}_2\}$$

donde ζ satisface la relación $\zeta^4+\zeta+1 = 0$. Usando esta relación, obtenemos la siguiente tabla para las potencias de ζ:

$$\begin{aligned}
\zeta^4 &= 1 + \zeta & \zeta^{10} &= 1 + \zeta + \zeta^2\\
\zeta^5 &= \zeta + \zeta^2 & \zeta^{11} &= \zeta + \zeta^2 + \zeta^3\\
\zeta^6 &= \zeta^2 + \zeta^3 & \zeta^{12} &= 1 + \zeta + \zeta^2 + \zeta^3\\
\zeta^7 &= 1 + \zeta + \zeta^3 & \zeta^{13} &= 1 + \zeta^2 + \zeta^3\\
\zeta^8 &= 1 + \zeta^2 & \zeta^{14} &= 1 + \zeta^3\\
\zeta^9 &= \zeta + \zeta^3 & \zeta^{15} &= 1
\end{aligned}$$

Ahora ζ es la 15-ésima raíz primitiva de la unidad en $\mathbb{F}_{2^4}$, y $p(x)$ es el polinomio minimal de ζ. Para obtener un código BCH de distancia designada $d = 7$, necesitamos los polinomios minimales de ζ^i para $i = 1, \ldots, 6$. Por el teorema 33, ζ, ζ^2, ζ^4 tienen el mismo polinomio minimal $p(x)$. Sea $q(x)$ el polinomio minimal de ζ^3, entonces $\zeta^3, \zeta^6, \zeta^{12}, \zeta^{24}, \ldots$ todos tienen el mismo polinomio minimal $q(x)$. Usando la relación $\zeta^{15} = 1$, vemos que las raíces de $q(x)$ son $\zeta^3, \zeta^6, \zeta^9, \zeta^{12}$. Por lo tanto

$$\begin{aligned}
q(x) &= (x - \zeta^3)(x - \zeta^6)(x - \zeta^9)(x - \zeta^{12})\\
&= x^4 - (\zeta^3 + \zeta^6 + \zeta^9 + \zeta^{12})x^3 + (\zeta^3 + \zeta^6 + \zeta^9 + \zeta^{12})x^2 - (\zeta^3 + \zeta^6 + \zeta^9 + \zeta^{12})x + 1\\
&= x^4 + x^3 + x^2 + x + 1
\end{aligned}$$

De manera similar el polinomio minimal $h(x)$ de ζ^5 tiene raíces ζ^5, ζ^{10}, así

$$\begin{aligned}
h(x) &= (x - \zeta^5)(x - \zeta^{10})\\
&= x^2 + x + 1
\end{aligned}$$

Por lo tanto el polinomio generador del código BCH deseado es

$$\begin{aligned}
g(x) &= lcm\{m_i(x) |\, i = 1, 2, 3, 4, 5, 6\}\\
&= p(x)q(x)h(x)\\
&= x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1
\end{aligned}$$

Por el teorema 24, el código cíclico C generado por $g(x)$ en $\mathbb{F}_2[15]_{15}$ tiene dimensión 5, por lo que C es un $[15, 5]$-código BCH primitivo en el sentido estrecho con distancia designada 7. Por lo tanto $d(C) \geq 7$. Ahora $g(x)$ es por si mismo un código polinomial y tiene 7 términos no nulos, así este es una palabra-código de peso 7. En consecuencia, por el teorema 15, $d(C) \leq 7$. Esto prueba que la distancia mínima de C es 7, por lo que C es un $[15, 5, 7]$-código.

Ejemplo 32. Con $n = 31$, $q = 2$, tenemos $m = 5$ y $2^5 - 1 = 31$. Sea ζ un elemento primitivo en $\mathbb{F}_{2^5}$, así ζ una 31-ésima raíz primitiva de la unidad. Sea $p(x)$ el polinomio minimal de ζ. Entonces ζ, ζ^2, ζ^4 tienen el mismo polinomio minimal $p(x)$. Sea $q(x)$ el polinomio minimal de ζ^3. Escribamos $g(x) = p(x)q(x)$. Entonces $g(x) = lcm\{m_i(x) | \, i = 1, 2, 3, 4\}$. Por lo tanto el código C con polinomio generador $g(x)$ en $\mathbb{F}_2[x]_{31}$ es un código de longitud 31 y distancia designada 5; Ahora $p(x)$ y $q(x)$ son polinomios mónicos de grado 5; por lo tanto el grado de $g(x)$ es 10. Por lo tanto la dimensión del código es 21, así C es un $[31, 21]$-código.

Ahora pasaremos a discutir la decodificación con códigos BCH. Explicamos en el capitulo 4 que par un código lineal en general preparamos una tabla de síndromes con el propósito de decodificar. Cuando un vector y es recibido, calculamos su síndrome $S(y) = yH^T$ y entonces revisamos la tabla de síndromes par encontrar el vector error e tal que $S(y) = S(e)$. Entonces y es decodificado como el vector transmitido $x = y - e$. Para un código BCH, no obstante, tenemos un método algebraico para encontrar el vector error e a partir del vector síndrome $S(y)$. A continuación presentaremos una breve descripción de este método.

TEOREMA PRINCIPAL. Sea C un código BCH de longitud n sobre un campo arbitrario finito $\mathbb{F}$ corrector de t errores. Si r es una palabra que tiene un vector de error e con peso $\leq t$, entonces r puede ser eficientemente decodificado.

Demostración. Sea C un código BCH sobre $F = \mathbb{F}_q$ de longitud n y distancia designada d. Sea H una matriz de tamaño $(d-1) \times n$ sobre $\mathbb{F}_{q^m}$ dado en el teorema 34. (Como se menciono anteriormente, H es una matriz cuasi-verificadora de paridad para C.) Usaremos esta matriz para definir el síndrome de un vector $a \in F^n$ como $S(a) = aH^T$. Escribiendo $a = (a_0, a_1, a_2, \ldots, a_{n-1})$ y $a(x) = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}$, tenemos

$$\begin{aligned} S(a) &= \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \end{bmatrix} \begin{bmatrix} 1 & \zeta^c & \zeta^{2c} & \cdots & \zeta^{(n-1)c} \\ 1 & \zeta^{c+1} & \zeta^{2(c+1)} & \cdots & \zeta^{(n-1)(c+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{c+d-2} & \zeta^{2(c+d-2)} & \cdots & \zeta^{(n-1)(c+d-2)} \end{bmatrix}^T \\ &= \begin{bmatrix} S_c & S_{c+1} & \cdots & S_{c+d-2} \end{bmatrix} \end{aligned}$$

donde

$$S_j = a_0 + a_1\zeta^j + \cdots + a_{n-1}\zeta^{(n-1)j} = a(\zeta^j)$$

para $j = c, c+1, \ldots, c+d-2$.

Ahora supongamos que una palabra-código $v \in C$ es transmitida y el vector recibido es $a = v + e$, donde e es el vector error. Entonces $S(e) = S(a)$. Sea $e = (e_0, e_1, \ldots, e_{n-1})$ y $e(x) = e_0 + e_1 x + \cdots + e_{n-1}x^{n-1}$. Sean $i_1, \ldots, i_r$ las posiciones en las que un error ha ocurrido. Entonces $e \neq 0$ si y sólo si $i \in I = \{i_1, \ldots, i_r\}$. Por lo tanto

$$e(x) = \sum_{i \in I} e_i x^i.$$

El código C puede corregir hasta t errores, donde $t = \lfloor (d-1)/2 \rfloor$. Así asumimos que $r \leq t$, esto es $2r < d$. Como $S(e) = S(a)$, tenemos $e(\zeta^j) = S_j$ para todo $j = c, c+1, \ldots, c+d-2$. De este modo las $2r$ incógnitas $i_1, \ldots, i_r$ y $e_{i_1}, \ldots, e_{i_r}$ satisfacen el siguiente sistema de $d-1$ ecuaciones lineales en $e_{i_1}, \ldots, e_{i_r}$:

$$\sum_{i \in I} e_i \zeta^{ji} = S_j \quad j = c, c+1, \ldots, c+d-2 \tag{1}$$

Primero obtenemos una solución para las posiciones de los errores $i_1, \ldots, i_r$. Definimos el polinomio localizador de errores $f(x)$ como

$$f(x) = \prod_{i \in I}(x - \zeta^i) = f_0 + f_1 x + \cdots + f_{r-1}x^{r-1} + x^r.$$

Como $f(\zeta^i) = 0$ para cada $i \in I$ tenemos

$$f_0 + f_1\zeta^i + \cdots + f_{r-1}\zeta^{(r-1)i} + \zeta^{ri} = 0$$

para cada $i \in I$. Multiplicando esta ecuación por $e_i\zeta^{ji}$, obtenemos

$$f_0 e_i\zeta^{ji} + f_1 e_i\zeta^{(j+1)i} + \cdots + f_{r-1}e_i\zeta^{(j+r-1)i} + e_i\zeta^{(j+r)i} = 0$$

para cada $i \in I$. Sumando estas r ecuaciones para $= i_1, \ldots, i_r$ y usando la relación (1) tenemos

$$f_0 S_j + f_1 S_{j+1} + \cdots + f_{r-1}S_{j+r-1} + S_{j+r} = 0$$

para cada $j = c, c+1, \ldots, c+d-1$. Así las r incógnitas $f_0, f_1, \ldots, f_{r-1}$ satisfacen el siguiente sistema $r \times r$ de ecuaciones lineales:

$$\begin{bmatrix} S_c & S_{c+1} & \cdots & S_{c+r-1} \\ S_{c+1} & S_{c+2} & \cdots & S_{c+r} \\ \vdots & \vdots & \ddots & \vdots \\ S_{c+r+1} & S_{c+r+2} & \cdots & S_{c+2r-2} \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} -S_{c+r} \\ -S_{c+r+1} \\ \vdots \\ -S_{c+2r-1} \end{bmatrix} \tag{2}$$

Si S denota la matriz de coeficiente en el anterior sistema lineal, se puede verificar por calculo directo que $S = VDV^T$, donde

$$V = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \zeta^{i_1} & \zeta^{i_2} & \cdots & \zeta^{i_r} \\ \vdots & \vdots & \ddots & \vdots \\ \zeta^{i_1(r-1)} & \zeta^{i_2(r-1)} & \ldots & \zeta^{i_r(r-1)} \end{vmatrix}, \quad D = \begin{vmatrix} e_{i_1}\zeta^{i_1 c} & 0 & \cdots & 0 \\ 0 & e_{i_2}\zeta^{i_2 c} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_{i_r}\zeta^{i_r c} \end{vmatrix}$$

Donde V es una matriz de Vandermonde, como Sea ζ una n-ésima raíz primitiva de la unidad en $\mathbb{F}_{q^m}$ y $i_1, \ldots, i_r$ son enteros diferentes de $\{0, 1, \ldots, n-1\}$, tenemos que $\zeta^{i_1}, \ldots, \zeta^{i_r}$ son todos diferentes. Por lo tanto el determinante de Vandermonde es no nulo. Además, $e_{i_1}, \ldots, e_{i_r}$ son todos no nulos y en consecuencia $D \neq 0$. Por lo tanto $\det S \neq 0$, y el sistema lineal (2) tiene solución única.

La solución única de (2) así obtenida nos da el polinomio localizador de errores

$$f(x) = f_0 + f_1 x + \cdots + f_{r-1}x^{r-1} + x^r.$$

Ahora encontremos las raíces de $f(x)$ probando $x = \zeta^i$, $i = 0, 1, \ldots$. Por la definición de $f(x)$, estas raíces son $\zeta^{i_1}, \ldots, \zeta^{i_r}$. Así obtenemos una única solución para las incógnitas $i_1, \ldots, i_r$. Si el código C es binario, entonces $e_{i_1}, \ldots, e_{i_r}$ son todos iguales a 1, así el error polinomial

$$e(x) = x^{i_1} + \cdots + x^{i_r}.$$

En el caso general no binario, encontramos $e_{i_1}, \ldots, e_{i_r}$ resolviendo el sistema de ecuaciones (1). Habiendo encontrado el vector error e, decodificamos el vector recibido a como la palabra código $v = a - e$.

Notemos que la matriz H es formalmente determinada por los números c, n, d. Para calcular el síndrome no necesitamos conocer el polinomio generador $g(x)$ del código BCH. Además por el teorema 32, si el código es sobre $\mathbb{F}_q$ entonces $S_q = (S_1)^q$. En particular, par un código binario, $S_2 = (S_1)^2$, $S_4 = (S_2)^2$, $S_6 = (S_3)^2$, y así en

adelante. Podemos calcular el síndrome con mayor facilidad empleando el algoritmo de la división. Si $p(x)$ es el polinomio minimal de ζ, entonces $S_1 = a(\zeta)$ puede ser obtenido encontrando el residuo de dividir $a(x)$ por $p(x)$ y entonces colocando $x = \zeta$ en el. En general, para encontrar S_j, dividimos $a(x^j)$ por $p(x)$ y encontrando el residuo.

Nota que si sólo un error ha ocurrido, digamos en la i-ésima posición, entonces $S(a)$ es igual a la i-ésima columna de H. Por lo tanto, si $S(a)$ es no nulo y no es igual a cualquier columna de H, entonces al menos dos errores han ocurrido. □

El siguiente ejemplo ilustra el proceso de decodificación.

Ejemplo 33. Supongamos que el $[15, 5, 7]$-código BCH del ejemplo 16 es usado y el vector

$$a = 110001001101000 \in \mathbb{F}_2^{15}$$

es recibido. Encontraremos el vector error y determinaremos la palabra-código correcta. entonces encontraremos la palabra del mensaje.

El polinomio $a(x)$ para el vector recibido $a = 110001001101000$ es

$$a(x) = 1 + x + x^5 + x^8 + x^9 + x^{11} \in \mathbb{F}_2[x]_{15}.$$

Por lo que los componentes S_i del vector síndrome $S(a)$ están dadas por

$$S_i = a(\zeta^i) = 1 + \zeta^i + \zeta^{5i} + \zeta^{8i} + \zeta^{9i} + \zeta^{11i}$$

para cada $i = 1, \ldots, 6 = d-1$. Usando las relaciones dadas en la tabla para potencias de ζ en el ejemplo 16, obtenemos

$$\begin{aligned}
S_1 &= 1 + \zeta + \zeta^5 + \zeta^8 + \zeta^9 + \zeta^{11} = \zeta^2 \\
S_2 &= (S_1)^2 = 1 + \zeta = \zeta^4 \\
S_3 &= 1 + \zeta^3 + \zeta^{15} + \zeta^{24} + \zeta^{27} + \zeta^{33} = 1 + \zeta^2 = \zeta^8 \\
S_4 &= (S_2)^2 = 1 + \zeta^2 = \zeta^8 \\
S_5 &= 1 + \zeta^5 + \zeta^{25} + \zeta^{40} + \zeta^{45} + \zeta^{55} = 1 \\
S_6 &= (S_3)^2 = \zeta
\end{aligned}$$

El código C corrige hasta 3 errores, por lo que asumimos que el numero de errores que han ocurrido en el vector recibido a es $r \leq 3$. Ahora $S(a)$ no es igual ninguna columna de H. Por lo tanto dos errores han ocurrido, así $r = 2$ o $r = 3$. Primero probemos $r = 3$; sea

$$f(x) = f_0 + f_1 x + f_2 x^2 + x^3$$

el polinomio localizador de errores. Entonces tenemos el siguiente sistema de ecuaciones para los coeficientes incógnitas f_0, f_1, f_2:

$$\begin{bmatrix} S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \\ S_3 & S_4 & S_5 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} -S_4 \\ -S_5 \\ -S_6 \end{bmatrix}$$

El sistema tiene única solución si y sólo si la matriz de coeficientes S es no singular. Ahora

$$\begin{aligned}
\det S &= \begin{vmatrix} \zeta^2 & \zeta^4 & \zeta^8 \\ \zeta^4 & \zeta^8 & \zeta^8 \\ \zeta^8 & \zeta^8 & 1 \end{vmatrix} \\
&= \zeta^{14} \begin{vmatrix} 1 & 1 & 1 \\ \zeta^2 & \zeta^4 & 1 \\ \zeta^6 & \zeta^4 & \zeta^7 \end{vmatrix} \\
&= \zeta^{14}(1 + \zeta^5 + \zeta^6 + \zeta^7) = 0
\end{aligned}$$

Por lo tanto r no puede ser igual a 3, así que $r = 2$. Entonces el polinomio localizador de errores es $f(x) = f_0 + f_1 x + x^2$, y el sistema lineal para f_0, f_1 es

$$\begin{bmatrix} S_1 & S_2 \\ S_2 & S_3 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \end{bmatrix} = \begin{bmatrix} -S_3 \\ -S_4 \end{bmatrix}$$

Ahora

$$\det S = \begin{vmatrix} \zeta^2 & \zeta^4 \\ \zeta^4 & \zeta^8 \end{vmatrix} = \zeta^{10} + \zeta^8 = \zeta.$$

Por lo tanto tenemos una solución única dada por

$$f_0 = \frac{S_3^2 - S_2 S_4}{\det S} = \frac{\zeta^{16} + \zeta^{12}}{\zeta} = \zeta^{12}$$
$$f_1 = \frac{S_2 S_3 + S_1 S_4}{\det S} = \frac{\zeta^{12} + \zeta^{10}}{\zeta} = \zeta^2$$

Así el polinomio localizador de errores es $f(x) = \zeta^{12} + \zeta^2 x + x^2$. Por prueba y error encontramos que una raíz es ζ^{13}, así la otra raíz es $f_0/\zeta^{13} = \zeta^{14}$. De esta manera las posiciones de los errores son 13 y 14, y el polinomio error es $e(x) = x^{13} + x^{14}$.

El polinomio corrector de código es

$$v(x) = a(x) - e(x) = 1 + x + x^5 + x^8 + x^9 + x^{11} + x^{13} + x^{14}$$

así $v = 110001001101011$. Para obtener la palabra mensaje, recordemos que para un código cíclico con polinomio generador $g(x)$, un polinomio mensaje $u(x)$ es decodificado como el polinomio código $v(x) = u(x)g(x)$. Así, en el presente caso,

$$\begin{aligned} u(x) &= \frac{v(x)}{g(x)} \\ &= \frac{1 + x + x^5 + x^8 + x^9 + x^{11} + x^{13} + x^{14}}{1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}} \\ &= x^4 + x^3 + x^2 + 1. \end{aligned}$$

Por lo tanto la palabra mensaje es 10111.

Mencionamos que para un código BCH no necesitamos conocer el polinomio generador si lo que se quiere es calcular el síndrome de un vector recibido y decodificarlo. El siguiente ejemplo ilustra esto.

Ejemplo 34. Sea C el $[31, 21]$-código BCH del ejemplo 32 y supongamos que el polinomio minimal de ζ es $p(x) = x^5 + x^2 + 1$. Decodificaremos el vector recibido

$$a = 1011000100100111000110111001011.$$

Usando la relación $\zeta^5 + \zeta^2 + 1 = 0$, calculamos las potencias de ζ y se obtiene la siguiente tabla:

$\zeta^5 = 1 + \zeta^2$	$\zeta^{14} = 1 + \zeta^2 + \zeta^3 + \zeta^4$	$\zeta^{23} = 1 + \zeta + \zeta^2 + \zeta^3$
$\zeta^6 = \zeta + \zeta^3$	$\zeta^{15} = 1 + \zeta + \zeta^2 + \zeta^3 + \zeta^4$	$\zeta^{24} = \zeta + \zeta^2 + \zeta^3 + \zeta^4$
$\zeta^7 = \zeta^2 + \zeta^4$	$\zeta^{16} = 1 + \zeta + \zeta^3 + \zeta^4$	$\zeta^{25} = 1 + \zeta^3 + \zeta^4$
$\zeta^8 = 1 + \zeta^2 + \zeta^3$	$\zeta^{17} = 1 + \zeta + \zeta^4$	$\zeta^{26} = 1 + \zeta + \zeta^2 + \zeta^4$
$\zeta^9 = \zeta + \zeta^3 + \zeta^4$	$\zeta^{18} = 1 + \zeta$	$\zeta^{27} = 1 + \zeta + \zeta^3$
$\zeta^{10} = 1 + \zeta^4$	$\zeta^{19} = \zeta + \zeta^2$	$\zeta^{28} = \zeta + \zeta^2 + \zeta^4$
$\zeta^{11} = 1 + \zeta + \zeta^2$	$\zeta^{20} = \zeta^2 + \zeta^3$	$\zeta^{29} = 1 + \zeta^3$
$\zeta^{12} = \zeta + \zeta^2 + \zeta^3$	$\zeta^{21} = \zeta^3 + \zeta^4$	$\zeta^{30} = \zeta + \zeta^4$
$\zeta^{13} = \zeta^2 + \zeta^3 + \zeta^4$	$\zeta^{22} = 1 + \zeta^2 + \zeta^4$	$\zeta^{31} = 1$

Si lo escribimos como polinomio, el mensaje recibido es

$$\begin{aligned} a(x) =& 1 + x^2 + x^3 + x^7 + x^{10} + x^{13} + x^{14} + x^{15} + x^{19} + x^{20} \\ &+ x^{22} + x^{23} + x^{24} + x^{27} + x^{29} + x^{30} \end{aligned}$$

Como el código C es de distancia designada 5, el síndrome de a es

$$S(a) = \begin{bmatrix} S_1 & S_2 & S_3 & S_4 \end{bmatrix}$$

donde $S_j = a(\zeta^j)$, $j = 1, 2, 3, 4$. Dividiendo $a(x)$ por $p(x)$, conseguimos el residuo x^3; Por lo tanto $S_1 = \zeta^3$. De manera similar $S_3 = \zeta + \zeta^2 + \zeta^3$. Usando la tabla de arriba, tenemos

$$\begin{aligned} S_1 &= a(\zeta) = \zeta^3 \\ S_2 &= S_1^2 = \zeta^6 \\ S_3 &= a(\zeta^3) = \zeta + \zeta^2 + \zeta^3 = \zeta^{12} \\ S_4 &= S_2^2 = \zeta^{12} \end{aligned}$$

Ahora $t = 2$; por lo que asumimos que el numero de errores será $r \leq 2$. Pero el vector síndrome no es igual a cualquier columna de H, por lo que, al menos 2 errores han ocurrido. Entonces $r = 2$. Sea el polinomio localizador de errores $f(x) = f_0 + f_1 x + x^2$. Los coeficientes f_0, f_1 satisfacen las ecuaciones

$$\begin{aligned} S_1 f_0 + S_2 f_1 &= S_3, \\ S_2 f_0 + S_3 f_1 &= S_4. \end{aligned}$$

Resolviendo estas ecuaciones, tenemos

$$\begin{aligned} f_0 &= \frac{\zeta^{18} + \zeta^{24}}{\zeta^{15} + \zeta^{12}} = \zeta^4, \\ f_1 &= \frac{\zeta^{18} + \zeta^{15}}{\zeta^{15} + \zeta^{12}} = \zeta^3. \end{aligned}$$

Por lo tanto el polinomio localizador de errores es $f(x) = \zeta^4 + \zeta^3 x + x^2$. Probando sucesivamente con $x = 1, \zeta, \zeta^2, \ldots$ encontramos que ζ^{10} es una raíz, entonces la otra raíz es $\zeta^4/\zeta^{10} = \zeta^{25}$. Por lo tanto el error polinomial es $e(x) = x^{10} + x^{25}$, con el que decodificamos $a(x)$ como

$$\begin{aligned} v(x) =& a(x) - e(x) \\ =& 1 + x^2 + x^3 + x^7 + x^{13} + x^{14} + x^{15} + x^{19} + x^{20} \\ &+ x^{22} + x^{23} + x^{24} + x^{25} + x^{27} + x^{29} + x^{30} \end{aligned}$$

La palabra-código corregida es $v = 101100010000011100011011110 1011$.

Ejemplo Comparativa. Consideremos un código binario BCH corrector de 2 errores de longitud 15. En esta caso el campo-alfabeto es $F = \mathbb{B}$. Sea α un elemento primitivo en alguna extensión de $\mathbb{B}$ de orden 15. Más precisamente la extensión resulta ser $\mathbb{B}[x]/\langle x^4 + x + 1\rangle$, luego este código, si lo vemos como un subespacio, $C \subseteq \mathbb{B}^{15}$, y si lo vemos como un ideal, $C \subseteq \mathbb{B}[x]/\langle x^{15} - 1\rangle$. Supongamos que recibimos la palabra

$$r = 001000100000000 = x^2 + x^6.$$

Paso 1. En este caso $t = 2$, luego el síndrome es $s(x) = s_0 + s_1x + s_2x^2 + s_3x^3$. Por (G) los coeficientes del síndrome se calculan de la siguiente forma:

$$\begin{aligned} s_0 &= r(\alpha) = \alpha^2 + \alpha^6 = \alpha^3, \\ s_1 &= r(\alpha^2) = \alpha^4 + \alpha^{12} = \alpha^6, \\ s_2 &= r(\alpha^3) = \alpha^6 + \alpha^{18} = \alpha^2, \\ s_3 &= r(\alpha^4) = \alpha^8 + \alpha^{24} = a^{12}. \end{aligned}$$

De donde, $s(x) = \alpha^3 + \alpha^6x + \alpha^2x^2 + \alpha^{12}x^3$.

Paso 2. Efectuando el algoritmo euclidiano con $a_0(x) = x^4$ y $a_1(x) = s(x)$,

$$\begin{aligned} x^4 &= s(x)(\alpha^3x + \alpha^8) + (\alpha^{13}x^2 + \alpha^8x + \alpha^{11}), \\ s(x) &= a_2(x)(\alpha^{14}x + \alpha^{14}) + \alpha^{12}, \end{aligned}$$

Notemos que $k = 3$ es el primer índice tal que $a_3(x) < 2 = t$, ahora necesitamos calcular $d = \big[u_3(0)\big]^{-1}$. Para esto, calculemos:

$$\begin{aligned} u_0(x) &= 0, \\ u_1(x) &= 1, \\ u_2(x) &= u_0 - q_1u_1 = \alpha^3x + \alpha^8, \\ u_3(x) &= u_1 - q_2u_2 = 1 + (\alpha^{14}x + \alpha^{14})(\alpha^3x + \alpha^8), \\ &= \alpha^2x^2 + \alpha^{12}x + \alpha^9. \end{aligned}$$

Luego $d = \big[u_3(0)\big]^{-1} = \alpha^{-9} = \alpha^6$. De donde, el polinomio localizador de error es:

$$\sigma(x) = d\,u_3(x) = \alpha^6(\alpha^2 + x^2 + \alpha^{12}x + \alpha^9) = \alpha^8x^2 + \alpha^3x + 1.$$

Notemos que $\sigma(x) = (1 - a_1x)(1 - a_2x)$ por (D).

Paso 3. Las raíces de $\sigma(x)$ son [por ensayo y error] α^9 y α^{13}. De ahí $a_1 = \alpha^{-9}$ y $a_2 = \alpha^{-13}$, es decir, $a_1 = \alpha^6$ y $a_2 = \alpha^2$. Por (C)

$$\begin{aligned} e_6 &= b_1 \text{ ya que } \alpha^6 = a_1, \\ e_2 &= b_2 \text{ ya que } \alpha^2 = a_2. \end{aligned}$$

Luego,

$$\begin{aligned} e &= e_0e_1e_2e_3e_4e_5e_6e_7e_8e_9e_{10}e_{11}e_{12}e_{13}e_{14}, \\ e &= 001000100000000. \end{aligned}$$

Así, la palabra código enviada fue 0.

Conclusión. La eficiencia del Algoritmo presentado en el presente proyecto está comprobada, pues como se puede ver el método convencional usa como herramienta el ALGORITMO EUCLIDIANO para el proceso de decodificación lo cual implica mayor requerimiento de procesos y bucles en la codificación del Algoritmo lo cual a mayor cantidad de palabras código a decodificar conllevaría a la saturación del algoritmo y del sistema.

Bibliografía

[1] T. W. Hungerford, *Abstract algebra*

[2] *Applied abstract algebra, Rings of Continuous Functions*, Springer Verlag, 1960.

[3] V. V. Prasolov, *Intuitive Topology*, American Mathematical Society, 1995.

[4] J. R. Weeks, *The Shape of Space*, 2nd ed. Marcel Dekker, 2002.

[5] D. S. Dummit & R. Foot, *Abstract Algebra*, third Edition, 2003.

[6] I. N. Herstein, *Álgebra abstracta*.

[7] J. Van Lint, *introduction to coding theory*.

[8] J. Gallian, *contemporary abstract algebra*.